



BACHELORARBEIT

Herr
Christian Bausch

**Berechnung der
 s, t -Zuverlässigkeit mit
Weglängenbeschränkung in
Netzwerken**

2013

BACHELORARBEIT

Berechnung der s, t -Zuverlässigkeit mit Weglängenbeschränkung in Netzwerken

Autor:

Christian Bausch

Studiengang:

Angewandte Mathematik

Seminargruppe:

MA10w1-B

Erstprüfer:

Prof. P. Tittmann

Zweitprüfer:

M.Sc. M. Dod

Mittweida, November 2013

Bibliografische Angaben

Bausch, Christian: Berechnung der s, t -Zuverlässigkeit mit Weglängenbeschränkung in Netzwerken, 73 Seiten, 42 Abbildungen, Hochschule Mittweida, University of Applied Sciences, Fakultät Mathematik/Naturwissenschaften/Informatik

Bachelorarbeit, 2013

Dieses Dokument ist urheberrechtlich geschützt.

Satz: \LaTeX

Referat

Diese Arbeit beschäftigt sich mit der Erstellung eines Verfahrens zur Berechnung des s, t - Zusammenhangspolynoms mit Weglängenbeschränkung in ungerichteten Netzwerken. Dieses Verfahren soll eine Baumzerlegung des Netzwerkes verwenden. Zudem werden einige Reduktionsmöglichkeiten vorgestellt.

I. Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	II
Vorwort	III
1 Grundlagen und Problemstellung	1
1.1 Grundlegende Definitionen	1
1.2 Einführung des Zusammenhangspolynoms $R_{s,t}(G, D)$	2
1.3 Komplexität	6
1.3.1 $D = 1$	6
1.3.2 $D = 2$	6
1.3.3 $D \geq 3$	9
2 Baumzerlegung	13
2.1 Einführung	13
2.1.1 Anordnung der Baumzerlegung	16
2.2 Die Verwendung einer Baumzerlegung	17
2.2.1 Beschreibung der Arbeitsweise mit Baumzerlegung	17
2.2.2 Partitionen als Informationsspeicher	20
2.2.3 Formulierung der vier Grundschrte	20
2.3 Erweiterung zur Berechnung des K -terminalen Zusammenhangspolynoms	28
2.3.1 Komplexität	31
2.4 Erweiterung zur Berechnung des s, t -Zusammenhangs	31
2.5 Komplexität zur Berechnung des s, t -Zusammenhangspolynoms	31
3 Berechnung der weglängenbeschränkten Zuverlässigkeit	33
3.1 Reduktionen	33
3.1.1 Schlingen-Reduktion	35
3.1.2 Brücken-Reduktion	36
3.1.3 Parallel-Reduktion	36
3.1.4 Weg-Reduktion	37
3.1.5 Artikulations-Reduktion	38
3.1.6 Inzidenz-Reduktion	38
3.1.7 Reduktion irrelevanter Kanten	39
3.1.8 Serien-Reduktion	43
3.1.9 Hinweis zur Anwendung	44
3.2 Einführende Idee	44
3.2.1 Ausgangsüberlegungen	44
3.2.2 Die Idee	45
3.2.3 Vorteile	45
3.2.4 Nachteile	46
3.3 Einführung der Abstandsmatrizen	48

3.4	Anwendung der Abstandsmatrizen in der Zuverlässigkeitsberechnung	50
3.4.1	Aktivierung eines Knotens v	50
3.4.2	Deaktivierung eines Knotens v	51
3.4.3	Untersuchung einer Kante $e = \{u, v\}$	53
3.4.4	Fusion zweier Teilbäume t_1, t_2	56
3.4.5	Behandlung der Knoten s und t	58
3.4.6	Vorzeitiges Entfernen von Zuständen	59
3.4.7	Vorzeitige Eliminierung von Zuständen durch Abstandsanalyse	59
3.5	Komplexität der beschränkten Zuverlässigkeitsberechnung	63
3.5.1	Speicheraufwand in Schritt i	63
3.5.2	Komplexität der vier Grundschritte	64
4	Auswertung und offene Probleme	67
	Literaturverzeichnis	69

II. Abbildungsverzeichnis

1.1	Probabilistischer Graph	4
1.2	Zulässiger Graph für $D = 2$, [Can03]	7
1.3	Probabilistischer Graph, [Can04]	10
1.4	Ausfallzustand, [Can04]	11
1.5	$R_{s,t}(\text{Baum}, 4, p) = p^4$	12
2.1	Baum mit Teilbaum T (rot) und K -Knoten g, s, t, v	13
2.2	Vollständiger Graph (blau) mit zwei Leitergraphen (rot)	14
2.3	Baumstruktur der Abbildung 2.2	14
2.4	Beispiel für einen Graphen und seine Baumzerlegung	15
2.5	Erweiterung von Abbildung 2.4 zur schönen Baumzerlegung	16
2.6	Darstellung von Wurzel, Vater und Kind	16
2.7	Entwicklung eines Binärbaumes	17
2.8	Orange: bereits deaktiviert; Rot: aktive Menge (Front); Blau: unbearbeitet	18
2.9	Ausgangsbeispiel für nachfolgende Grundschritte mit Beispielpolynom	21
2.10	Aktivierung eines Knotens	22
2.11	Umsetzung der Deaktivierung im Graph	22
2.12	Umsetzung der Deaktivierung in den Zuständen	24
2.13	Umsetzung der Untersuchung im Graph für die Kante $\{2, 3\}$	25
2.14	Umsetzung der Untersuchung der Kante $\{2, 3\}$ in den Zuständen	26
2.15	Umsetzung der Fusion des roten und des grünen Teilbaums	27
3.1	Entfernung des Knotens 2 aus der Baumzerlegung	34
3.2	Umsetzung der Kantenkontraktion	34
3.3	Umsetzung der Kontraktion in der Baumzerlegung	35
3.4	Schlingen-Reduktion	36
3.5	Brücken-Reduktion	36
3.6	Parallel-Reduktion	36
3.7	Weg-Reduktion	37
3.8	Artikulations-Reduktion	38

3.9	Inzidenz-Reduktion	39
3.10	Reduktion der irrelevanten Kante im Diamantgraph	40
3.11	Beispiel für eine doppelt abgelaufene Kante (rot)	42
3.12	Serien-Reduktion	43
3.13	Weg (schwarz) über zwei Blöcke (rot, orange für deaktivierte Knoten)	47
3.14	Problem fehlerhafter kürzester Wege	47
3.15	Beispiel einer Abstandsmatrix; G besteht aus den roten Knoten	48
3.16	Schwarz: Komponente/Block 1; Grün: Komponente/Block 2; Partition: $\text{su}v r$	49
3.17	Deaktivierung des Knotens s	50
3.18	Umsetzung für u, v in verschiedenen Blöcken	54
3.19	Umsetzung für u, v im selben Block	55
3.20	$d(x, u)$ und $d(y, u)$ falsch, damit sind l_1, l_2 falsch	55
3.21	Interpretation der Fusion als Vereinigung zweier Teilgraphen	57
3.22	Graph der Abstandsanalyse	61

III. Vorwort

Man kann behaupten, dass unsere gesamte moderne Gesellschaft vernetzt ist. Pendler nutzen das Verkehrsnetz, um als Arbeitnehmer ihre Rolle im Produktionsnetz erfüllen zu können. Facebook-User nutzen das Social Network, um ihre sozialen Kontakte zu pflegen.

Computer-, Kommunikations- und andere Netzwerke spielen eine zunehmend größere Rolle für die Arbeit und das Funktionieren von Staaten, Unternehmen und Privatpersonen. Oft gilt es, Beschränkungen einzuhalten wie zeitliche Limits, Ressourcenkapazitäten oder Entfernungen. Störungen führen dabei oft zu erheblichen Umsatzeinbußen, Informations- oder Zeitverlusten. Daher ist man in unserer heutigen Zeit sehr daran interessiert, möglichst ausfallresistente Netzwerke zu gestalten.

So kommen beispielsweise immer wieder Zugverbindungen zum Erliegen, weil die Züge auf Grund des Wetters (zum Beispiel im Winter) nicht fahren können. Ein anderes Beispiel liefert das hohe Aufkommen an internetgebundenen multimedialen Mehrzweckgeräten wie Smartphones oder Tablet-PCs. Die Nutzer haben heutzutage einen hohen Erwartungsanspruch an die Sicherstellung eines störungsfreien Internetzuganges. Ein letztes für diese Arbeit relevantes Beispiel liefert die vom Staat angestrebte Einführung von einer Million Elektroautos auf deutschen Straßen. Auf Grund der heutigen Akkulaufzeiten sind ihre Fahrstrecken noch auf 150 km beschränkt. Um den Kauf eines solchen Autos dennoch attraktiv zu gestalten, muss auf unseren Straßen ein Netz von Ladestationen derart verteilt sein, dass auch bei einem Stau oder gesperrten Straßen wenigstens eine solche Station in erreichbarer Nähe liegt. Netzwerke müssen also so geplant werden, dass die Wahrscheinlichkeit für den Ausfall des Systems oder bestimmter Funktionsmerkmale gering ist.

Traditionellerweise modelliert man Netzwerke als Graphen, indem kommunikationsintensive Punkte wie Personen oder Computer als Knoten angesehen werden und die Kommunikation zwischen zwei derartigen Punkten als Kanten. Mit Hilfe der Graphentheorie ist es möglich, eine Vielzahl von Antworten auf unterschiedliche Fragen in Bezug auf die Zuverlässigkeit in Netzwerken zu liefern. Die Graphentheorie ist ein Teilgebiet der Mathematik, das die Eigenschaften von Graphen und ihre Beziehungen zueinander untersucht. Seit den 70-er Jahren wurden vor allem für ungerichtete Netzwerke tiefgehende Untersuchungen über die Berechnung von Zuverlässigkeitsmaßen angestellt.

Das bisher bekannte Berechnungsverfahren des s, t -Zusammenhangspolynoms unter Verwendung einer Baumzerlegung wird in dieser Arbeit aufgegriffen und verändert. Das Ziel soll sein, die Beschränkung der Weglänge des Graphen in das Verfahren zu integrieren. Zu diesem speziellen Fall existiert bisher keine Literatur. Es ist anzunehmen, dass er entweder noch nicht untersucht wurde, oder dass sich eine Lösung auf Basis einer Baumzerlegung als zu ineffizient gestaltete, so dass man der Sache nicht weiter nachging.

Um eine einheitliche Basis zu schaffen, vermittelt Kapitel 2 die graphentheoretischen

Grundlagen und bietet eine Übersicht über die verschiedenen Zuverlässigkeitsprobleme und ihrer Komplexität. Anschließend ist in Kapitel 3 das Berechnungsverfahren zur Ermittlung des Zuverlässigkeitspolynoms vorgestellt. Dazu wird die Baumzerlegung eingeführt und das Verfahren für die Anwendung auf den K -terminalen und s, t -terminalen Fall erweitert. In Kapitel 4 folgt der Hauptschwerpunkt dieser Arbeit, nach einem kurzen Einblick in verschiedene Reduktionen des Graphen zur Beschleunigung der Berechnung der Zuverlässigkeit. Von anfänglichen Überlegungen ausgehend, wird beschrieben, wie die Berechnungsmethode des vorherigen Kapitels modifiziert werden kann, so dass man das weglängenbeschränkte s, t -Zusammenhangspolynom unter Verwendung einer Baumzerlegung erhält. Das letzte Kapitel fasst die Ergebnisse dieser Arbeit zusammen und bietet zudem einen Ausblick auf die offenen Probleme dieser Methode.

1 Grundlagen und Problemstellung

Dieses Kapitel dient der Einführung der wichtigsten Definitionen der Arbeit.

1.1 Grundlegende Definitionen

Definition 1.1.1 (Graph)

Ein **Graph** $G = (V, E)$ ist ein geordnetes Paar, bestehend aus einer **Knotenmenge** $V \neq \emptyset$ (Elemente: **Knoten** (im engl.: vertices)) und einer **Kantenmenge** E (Elemente: **Kanten** (engl.: edges)), wobei mittels einer Inzidenzfunktion $\varphi : E \rightarrow \binom{V}{2}$ jeder Kante $e \in E$ zwei Endknoten aus V zugeordnet werden.

Bestehen die Endknoten einer Kante aus demselben Knoten nennt man diese **Schlinge**. Besitzt der Graph keine Kanten, so bezeichnet man ihn als **leer**.

Existiert eine Kante $e = \{v_1, v_2\}$, so sagt man, dass die Kante e die Knoten v_1 und v_2 verbindet.

Definition 1.1.2 (Digraph)

Ein Graph $G = (V, E)$ wird **gerichteter Graph** oder **Digraph** genannt, wenn die Elemente e, f der Kantenmenge geordnet sind, also

$$e = (1, 2) \neq (2, 1) = f.$$

Gerichtete Kanten werden oftmals auch als **Bögen** (engl.: arcs) bezeichnet, die Knoten als **Ecken** (im engl.: nodes). Man sagt, dass die gerichtete Kante e von 1 nach 2 geht. Einen ungerichteten Graphen nennt man auch **gewöhnlich**.

Definition 1.1.3 (Kantengewichteter Graph)

Ein Graph heißt **kantengewichtet** (kurz: **gewichtet**), wenn jeder Kante ein Kantengewicht $weight$ (in diesem Zusammenhang Länge) zugeordnet wird:

$$weight : E \rightarrow \mathbb{R}_+ := \{x \in \mathbb{R} : x \geq 0\}$$

Besitzt der Graph keine Kantengewichte, so nennt man ihn **ungewichtet**.

Ein ungerichteter, ungewichteter Graph ohne Schlingen oder mehrfachen (**parallelen**) Kanten zwischen den Knoten wird als **schlicht** bezeichnet.

Definition 1.1.4 (Beziehungen zwischen Knoten)

Gegeben ist ein gewöhnlicher Graph $G = (V, E)$. Man sagt, eine Kante e ist mit dem Knoten v **inzident**, wenn v einer der Endknoten von e ist.

Der **Grad eines Knotens** $\deg(v)$ (auch **Knotengrad** genannt) ist die Anzahl der zu v inzidenten Kanten (Schlingen werden doppelt gezählt).

Zwei durch eine Kante e verbundene Knoten werden **adjazent** oder **benachbart** genannt.

Die **Nachbarschaft** $N(v)$ von v im Graphen G bildet die Menge aller zu v adjazenten Knoten.

$$N(v) = \{u \in V \mid \{u, v\} \in E\}$$

Bemerkung 1.1.1

Sei $V' \subseteq V$, dann sind in der Menge $N(V')$ alle Knoten enthalten, die adjazent zu den Knoten aus V' sind.

1.2 Einführung des Zusammenhangspolynoms

$$R_{s,t}(G, D)$$

Nach der Einführung der wichtigsten Begriffe soll die Zusammenhangswahrscheinlichkeit schrittweise definiert werden. Dazu wird der Zusammenhang im Graphen erklärt, der die Verbindung von Knoten im Graphen beschreibt und danach die Wahrscheinlichkeit eingeführt.

Definition 1.2.1 (Weg)

Gegeben sei ein gewöhnlicher Graph $G = (V, E)$. Ein **Weg** P ist eine alternierende Folge von Knoten und Kanten

$$P = (v_1, \{v_1, v_2\}, v_2, \dots, v_i, \{v_i, v_{i+1}\}, v_{i+1}, \dots, v_j)$$

mit $v_i \in V$, $\{v_i, v_{i+1}\} \in E$, $i \in \{1, \dots, j-1\}$, wobei kein Knoten und keine Kante mehrfach vorkommt. Dabei werden v_1, v_j als **Endknoten** des Weges bezeichnet.

Bemerkung 1.2.1

Ein Weg in einem schlichten Graphen lässt sich durch eine bloße Kanten- oder Knotenfolge eindeutig charakterisieren. Des Weiteren soll das Kürzel s, t -Weg für einen Weg in einem Graphen G mit Endknoten s und t stehen.

Ein für diese Arbeit relevantes Wegmaß bildet die Länge:

Definition 1.2.2 (Weglänge)

Gegeben sei ein gewöhnlicher Graph $G = (V, E)$. Die **Länge** $l(P)$ eines Weges bezeichnet die Summe der Kantengewichte in P .

Der **Abstand** $d(s, t)$ zweier Knoten s und t im Graphen G beschreibt die Länge eines kürzesten s, t -Weges:

$$d(s, t) = \min_{P_i \in P_{s,t}} (l(P_i))$$

Dabei stellt $P_{s,t}$ die Menge aller s, t -Wege in G dar.

Bemerkung 1.2.2

Der Abstand zweier adjazenter Knoten in einem ungewichteten Graphen beträgt 1.

Untersucht werden in dieser Arbeit Wege, die eine vorgegebene Länge nicht überschreiten:

Definition 1.2.3 (Weglängenbeschränkung)

Gegeben sei ein gewöhnlicher Graph $G = (V, E)$. Unter der **Weglängenbeschränkung** D versteht man eine Zahl $D \in \mathbb{R}_+$, die in $P_{s,t}$ nur Wege der Länge kleiner gleich D zulässt. Die so entstandene neue Wegmenge wird mit $P_{s,t}(D)$ bezeichnet. Ist der Graph ungewichtet, so gilt:

$$D \in \mathbb{N}, 1 \leq D \leq n - 1$$

Jeder s, t -Weg der Länge kleiner gleich D heißt **D-Weg**.

Alle nötigen Definitionen wurden angegeben, um den Zusammenhang zu definieren:

Definition 1.2.4 (Zusammenhängender Graph)

Ein gewöhnlicher Graph $G = (V, E)$ heißt **zusammenhängend**, wenn zwischen je zwei Knoten ein Weg existiert.

Ein Graph $G' = (V', E')$ mit $V' \subseteq V$ und $E' \subseteq E$ nennt man **Untergraph** von G . Maximal zusammenhängende Untergraphen des Graphen sind **Komponenten**.

Nun können Kantenwahrscheinlichkeiten durch eine Erweiterung der Graphendefinition eingeführt werden:

Definition 1.2.5 (Probabilistischer Graph)

Ein **probabilistischer Graph** $G = (V, E, Pr(E))$ ist ein Graph mit einer Wahrscheinlichkeitsfunktion $Pr : E \rightarrow [0, 1]$, die jeder Kante e eine Wahrscheinlichkeit p_e zuordnet (siehe Abbildung 1.1).

Mit p_e bezeichnet man dabei die **Intaktheitswahrscheinlichkeit**, also die Wahrscheinlichkeit des Ereignisses, dass die Kante e intakt ist.

Die Wahrscheinlichkeit für das Eintreten des Gegenereignisses, die **Ausfallwahrscheinlichkeit**, wird mit $q_e = 1 - p_e$ beschrieben. Dabei fallen die Kanten stochastisch unabhängig voneinander aus.

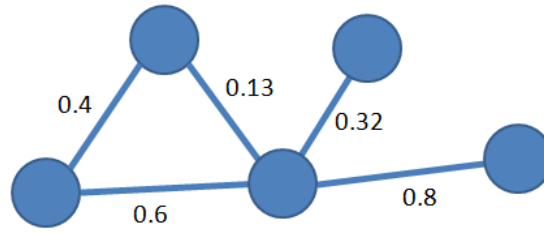


Abbildung 1.1: Probabilistischer Graph

Definition 1.2.6 (Stichprobenraum)

Gegeben sei ein ungerichteter probabilistischer Graph $G = (V, E, Pr(E))$. Der **Stichprobenraum** Ω bezeichnet die Menge aller Teilmengen der Kantenmenge E (kurz: Ω ist die Potenzmenge von E : $\Omega = 2^E$).

Ein Element H von Ω wird **Zustand** genannt.

Da die Knotenmenge unverändert erhalten bleibt, stellt Ω in Kombination mit V die Menge aller aufspannenden Untergraphen von G dar (Graphen mit $G' = (V, E'E)$, $E' \subseteq E$). Analog steht H mit V für genau einen möglichen Untergraphen von G . Demzufolge wird ein Zustand als das Ereignis angesehen, dass alle in H befindlichen Kanten von G intakt und die restlichen ausgefallen sind.

Die Wahrscheinlichkeit für das Eintreten von H und somit der Wahrscheinlichkeit, dass ein derartiger Untergraph entsteht, lautet:

$$Pr(H) = \prod_{e \in H} p_e \prod_{f \notin H} q_f$$

Zur Vereinfachung legen wir fest, dass alle Kanten dieselbe Wahrscheinlichkeit besitzen. Es gilt also $p = p_e$, $\forall e \in E$. Da p eine Variable ist, bildet $Pr(H)$ das sogenannte **Zustandspolynom**.

Definition 1.2.7 (Zusammenhangspolynom)

Die Summe aller Zustandspolynome, deren zugrundeliegender Graph H zusammenhängend ist, bezeichnet man als **Zusammenhangspolynom** $R(G)$. Besitzen alle Kanten dieselbe Ausfallwahrscheinlichkeit $1 - p$, so schreibt man $R(G, p)$. Dies gilt auch für alle weiteren Polynome.

Obiges Polynom ist ein Spezialfall des allgemeineren **K-terminalen Zusammenhangspolynoms** $R_K(G, p)$. Dabei ist $K \subseteq V$ eine Knotenteilmenge der Graphenknoten von G und das Interesse liegt im Zusammenhang der K -Knoten. Es wird jedoch ein anderer Spezialfall als $K = V$ untersucht: $K = \{s, t\}$

Diese Arbeit dient der Erforschung des Zusammenhangs zweier speziell ausgewiesener Knoten s und t . Die entsprechende Zuverlässigkeit wird durch das **s, t-Zusammenhangspolynom** $R_{s,t}(G, p)$ angegeben (engl.: *two-terminal reliability polynomial*). Analog

zum oben definierten Zusammenhangspolynom geht es aus den Polynomen derjenigen Zustände hervor, die zum Zusammenhang von s und t führen. Demzufolge treten in der Summe auch Polynome auf, deren Zustände den Graphen in mehrere Komponenten zerfallen lassen. Wichtig ist allein, dass ein Weg zwischen s und t existiert.

Definition 1.2.8 (Zulässiger Zustand)

H heißt **zulässig**, wenn der durch H entstandene Untergraph von G einen D -Weg enthält. Die **Menge aller zulässigen Zustände** bildet

$$O_{s,t}^D(E) = \{H \in \Omega : H \text{ ist zulässig}\}.$$

Die **Menge aller nicht zulässigen Zustände** bildet

$$\overline{O}_{s,t}^D(E) = \{H \in \Omega : H \text{ ist nicht zulässig}\}.$$

Ein zulässiger Zustand wird **Minimalweg** genannt, wenn $H - \{e_i\} \notin O_{s,t}^D(E)$, $\forall e_i \in H$. Damit enthält H bei Entfernung einer beliebigen Kante keinen D -Weg mehr und ist kein zulässiger Zustand.

Eine weitere Einschränkung ergibt sich aus der Weglängenbeschränkung D . Durch sie werden alle Zustandspolynome aus der Summe des s, t -Zusammenhangspolynoms entfernt, deren Graphen keinen D -Weg enthalten:

Definition 1.2.9 (Zusammenhangspolynom DC2RP)

Die Summe aller Zustandspolynome, mit zum s, t -Zusammenhang führenden zulässigen Zuständen, nennt man **weglängenbeschränktes s, t -Zusammenhangspolynom** $R_{s,t}(G, p, D)$ (engl.: diameter constrained two-terminal reliability polynomial, kurz **DC2RP**). Die damit verbundene Wahrscheinlichkeit bezeichnet man auch als **weglängenbeschränkte s, t -Zuverlässigkeit** (engl.: diameter constrained two-terminal-reliability). Es gilt:

$$R_{s,t}(G, D) = Pr\left(O_{s,t}^D(E)\right) = \sum_{H \in O_{s,t}^D(E)} \prod_{e \in H} p_e \prod_{f \notin H} q_f \quad (1.1)$$

bzw.

$$R_{s,t}(G, p, D) = Pr\left(O_{s,t}^D(E)\right) = \sum_{H \in O_{s,t}^D(E)} p^{|H|} q^{|V|-|H|} \quad (1.2)$$

Sowohl (1.1) als auch (1.2) lassen sich über das Gegenereignis, die unzulässigen Zustände, beschreiben. Auf Grund der Trivialität geben wir nur die Wahrscheinlichkeit des Gegenereignisses für (1.2) an:

$$R_{s,t}(G, p, D) = 1 - \sum_{H \in \overline{O}_{s,t}^D(E)} \prod_{e \in H} p^{|H|} q^{|V|-|H|}$$

Bemerkung 1.2.3

Den Begriff Zuverlässigkeit gebrauchen wir in dieser Arbeit oft als Synonym für das Zusammenhangspolynom. Da der Kontext klar ist, werden zumeist „weglängenbeschränkt“ und „ s, t “ weggelassen.

Damit ist das DC2RP als gesuchte Zuverlässigkeit definiert.

1.3 Komplexität

Dieser Abschnitt beschäftigt sich mit der Komplexität des DC2RP. Dazu werden nachfolgend die von Cancela und Petingi ([Can01], [Can03]) gefundenen Beweise verwendet.

Satz 1.3.1 (Cancela, Petingi)

Die Probleme der Berechnungen der Zuverlässigkeiten $R_K(G, D)$, $R_V(G, D)$ und $R_{s,t}(G, D)$ sind NP-schwer.

Es muss jedoch gesagt werden, dass das Problem des DC2RP nur bei festem $D > 2$ NP-schwer ist.

1.3.1 $D = 1$

Bei diesem trivialen Fall sind die einzigen Wege, die die erlaubte Länge von 1 nicht überschreiten, die k Kanten ($k \in \mathbb{N}$), die zur Adjazenz von s und t führen. Die Wahrscheinlichkeit, dass mindestens eine dieser Kanten intakt ist, wird durch die Wahrscheinlichkeit des Gegenereignisses beschrieben, dass alle Kanten ausfallen. Es gilt also:

$$R_{s,t}(G, p, 1) = 1 - \prod_{e \in P_{s,t}(1)} q_e$$

Besitzen alle k Kanten die gleiche Wahrscheinlichkeit p , so ist

$$R_{s,t}(G, p, 1) = 1 - q^k$$

1.3.2 $D = 2$

Durch die Weglängenbeschränkung $D = 2$ enthält $P_{s,t}(2)$ neben den Wegen des vorherigen Spezialfalles zusätzlich alle aus zwei Kanten bestehenden Wege. Diese zeichnen sich dadurch aus, dass s und t zu jeweils einer Kante inzident sind und es einen Knoten

$i \in V$ gibt, der zu beiden Kanten inzident ist. Damit gilt:

$$P_{s,t}(2) = \left\{ \{s,t\} \right\} \cup \left\{ \{s,i\}, \{i,t\} \mid i \in V \setminus \{s,t\} ; \{s,i\}, \{i,t\} \in E \right\}$$

Um das DC2RP zu berechnen, verwendet man die Methode der disjunkten Produkte. Zur Vereinfachung werden parallele Kanten zu einer einzigen Kante reduziert (siehe dazu Kapitel 3.1.3). Zudem wird angenommen, dass zwischen je zwei Knoten $i, j \in V$ eine Kante $\{i, j\} \in E$ existiert. Sollte es diese Kante nicht geben, so wird sie mit Wahrscheinlichkeit 0 eingeführt. Wir erhalten einen Graphen mit dem Aussehen wie in Abbildung 1.2.

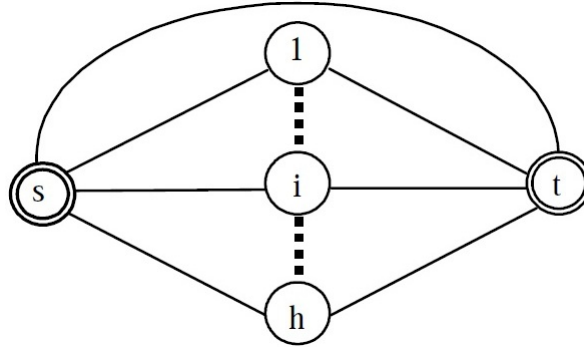


Abbildung 1.2: Zulässiger Graph für $D = 2$, [Can03]

Definition 1.3.1 (Wegwahrscheinlichkeit)

Gegeben sei ein ungerichteter probabilistischer Graph $G = (V, E, Pr(E))$ mit ausgezeichneten Knoten s und t . Die **Wegwahrscheinlichkeit** $Pr(E(P))$ ist die Wahrscheinlichkeit des Ereignisses, dass alle Kanten, die den D -Weg $P \in P_{s,t}(D)$ in G bilden, intakt sind:

$$Pr(E(P)) = \prod_{e \in P} p_e$$

Als nächstes bestimmt man die Wege in $P_{s,t}(2)$ und gibt ihnen laut Cancela und Petingi durch willkürliche Verteilung von Indizes eine Ordnung. Man sagt, dass $P_1 < P_2$, wenn P_1 vor P_2 kommt. Die Wahrscheinlichkeit, dass mindestens einer dieser Wege intakt ist, lautet:

$$R_{s,t}(G, 2) = Pr\left(\bigcup_{P_i \in P_{s,t}(2)} E(P_i)\right)$$

Das Ereignis, dass speziell P_i intakt ist, wird als das Ereignis dargestellt, dass alle vorherigen Wege ausgefallen sind und P_i intakt ist:

$$R_{s,t}(G, 2) = Pr\left(\bigcup_{P_i \in P_{s,t}(2)} \left(E(P_i) \cap \left(\bigcup_{P_j < P_i} \overline{E(P_j)}\right)\right)\right)$$

wobei $\overline{E(P_j)}$ das Gegenereignis zu $E(P_j)$ darstellt.

Es ist leicht einzusehen, dass die Ereignisse $E(P_i) \cap \left(\bigcup_{P_j < P_i} \overline{E(P_j)} \right)$ disjunkt sind, da die Kanten stochastisch unabhängig ausfallen.

$$R_{s,t}(G, 2) = \sum_{P_i \in P_{s,t}(2)} Pr \left(E(P_i) \cap \left(\bigcap_{P_j < P_i} \overline{E(P_j)} \right) \right)$$

Da es weiterhin keine Kanten gibt, die zwischen den Pfaden wiederholt auftreten, sind auch die Ereignisse, die in jedem Summanden auftauchen, unabhängig:

$$R_{s,t}(G, 2) = \sum_{P_i \in P_{s,t}(2)} Pr(E(P_i)) \prod_{P_j < P_i} Pr(\overline{E(P_j)})$$

Jedes Ereignis $E(P_i)$ ist der Durchschnitt der unabhängigen Ereignisse (beide Kanten sind intakt). Damit kann man die Terme $Pr(E(P_i))$ durch die Wahrscheinlichkeit ersetzen, dass beide Kanten intakt sind:

$$R_{s,t}(G, 2) = p_{\{s,t\}} + (1 - p_{\{s,t\}}) \sum_{i \in V \setminus \{s,t\}} p_{\{s,i\}} p_{\{i,t\}} \prod_{j < i, j \in V \setminus \{s,t\}} (1 - p_{\{s,j\}} p_{\{j,t\}})$$

Der erste Summand und der erste Faktor des darauf folgenden Produktes sind die Wahrscheinlichkeiten, dass die Kante zwischen s und t intakt bzw. nicht intakt ist. Bei den Indizes wurde auf Knoten gewechselt, denn diese Darstellung der Kanten erscheint im Sinne der Nachvollziehbarkeit angebrachter.

Da lediglich alle Kantenwahrscheinlichkeiten eingesetzt werden müssen, liegt die Laufzeit der Berechnung in $O(m)$.

Angenommen alle Kanten besitzen dieselbe Intaktheitswahrscheinlichkeit p , so lässt sich die Formel wie folgt schreiben ([Can03]):

$$\begin{aligned} R_{s,t}(G, p, 2) &= p + (1 - p) \sum_{i \in V \setminus \{s,t\}} p^2 \prod_{j < i, j \in V \setminus \{s,t\}} (1 - p^2) = p + (1 - p) p^2 \sum_{j=0}^{h-1} (1 - p^2)^j \\ &= p + (1 - p) p^2 \frac{1 - (1 - p^2)^h}{1 - (1 - p^2)} = 1 - (1 - p)(1 - p^2)^h \end{aligned} \quad (1.3)$$

wobei nur die Knoten $1, \dots, h$ betrachtet werden, die auf einem Weg der Länge 2 liegen.

Damit wurden die Fälle betrachtet, die seitens der Weglängenbeschränkung zu einer polynomialen Laufzeit in der Berechnung führen.

1.3.3 $D \geq 3$

Satz 1.3.2 (Cancela, Petingi)

Das Problem der Berechnung der Zuverlässigkeit $R_{s,t}(G, D)$ mit festem $D \geq 3$ ist NP-schwer ([Can01], [Can03]).

Für den Beweis seien folgende Definitionen gegeben:

Definition 1.3.2 (Knotenüberdeckende Menge)

Gegeben sei ein ungerichteter, probabilistischer, bipartiter Graph $G = (V, E, Pr(E))$ mit ausgezeichneten Knoten s und t . Seien X und Y Klassen der Bipartition von V . Eine **knotenüberdeckende Menge** ist eine Knotenmenge

$$C = C_X \cup C_Y$$

mit $C_X \subseteq X$ und $C_Y \subseteq Y$, so dass jede Kante von E mindestens einen Endknoten in C besitzt.

Definition 1.3.3 (Anzahl der Fehlerzustände)

Gegeben sei ein ungerichteter, probabilistischer, bipartiter Graph $G = (V, E, Pr(E))$ mit ausgezeichneten Knoten s und t und einer Weglängenbeschränkung D . Sei G' ein Untergraph von G und P ein D -Weg in G' .

Dann ist $F_i(G', \{s, t\}, D)$ die **Anzahl der Fehlerzustände** von G' mit $|E| + |E(P)| + i$ Kanten.

Beweis:

Cancela und Petingi haben in [Can04] gezeigt, dass die Bestimmung der verschiedenen Zustände eines ungerichteten Graphen mit Terminalknoten s und t und festem Parameter $D \geq 3$ in polynomialer Zeit Turing-reduzierbar auf die Bestimmung der Zahl der Knotenüberdeckungen eines bipartiten Graphen ist. In diesem Beweis wird sich darauf gestützt, dass Provan und Ball bereits 1983 die #P-Vollständigkeit des Knotenüberdeckungsproblems gezeigt haben ([Bal83]). Damit wäre die Bestimmung der Anzahl der Zustände #P-vollständig und die Berechnung der DC2R wegen der Ermittlung dieser Zustände NP-schwer.

Für den Beweis wird ein **probabilistischer Graph** $G' = (V', E', P(E'))$ (siehe Abbildung 1.3) konstruiert:

Sei $D = 3 + d$ und sei $P = (V(P), E(P))$ ein Weg über $d + 1$ Knoten ($V(P) = \{s, s_1, \dots, s_d\}$ und $E(P) = \{(s, s_1), (s_1, s_2), \dots, (s_{d-1}, s_d)\}$). Im Falle $d = 0$ besteht der Weg P aus einem einzigen Knoten s .

Man konstruiert den probabilistischen Graphen $G' = (V', E', P(E'))$ aus G , indem der Weg P und der Knoten t mit G verbunden werden:

Die Knotenmenge V' bestehe aus V , $V(P)$ und t . Die Kantenmenge E' bestehe aus

E , $E(P)$ und den folgenden Kantenmengen

$E_x = \{(s_d, x) : x \in X\}$ (die verbindenden Kanten vom letzten Knoten des Weges P mit den Knoten der Bipartitionsklasse X) und

$E_y = \{(t, y) : y \in Y\}$ (die verbindenden Kanten der anderen Bipartitionsklasse Y mit t) (siehe Abbildung 1.3).

Es gelte: $(p_e = 1 | e \in E \vee e \in E(P))$ ansonsten sei $p_e = \frac{1}{2}$

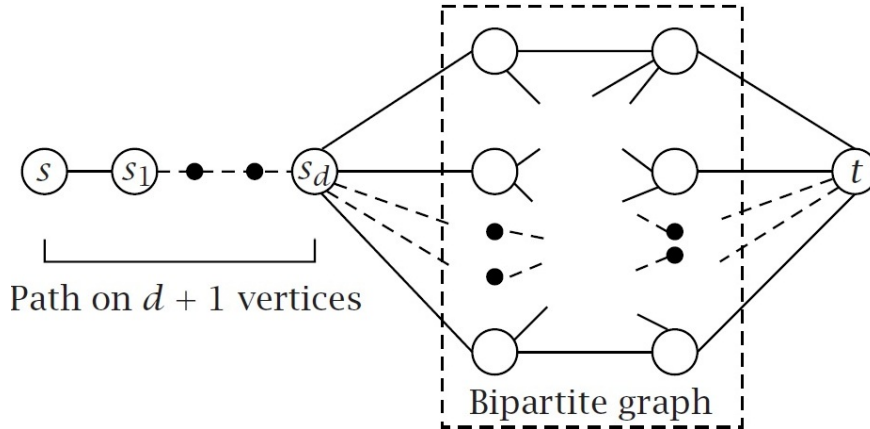


Abbildung 1.3: Probabilistischer Graph, [Can04]

Da D konstant, ist diese Transformation polynomial in der Größe der Eingabe des bipartiten Graphen. Damit ist der Schritt der Konstruktion beendet.

Um es noch einmal zusammenzufassen:

Ein Zustand H von G' besteht aus:

- dem bipartiten Graphen G
- dem Weg P
- dem Knoten s_d aus P und eventuellen Kanten zu Knoten aus X
- dem Knoten t und eventuellen Kanten zu Knoten aus Y

Der nächste Schritt widmet sich der Erstellung einer Bijektion $Z : W \mapsto \overline{O}_{s,t}^D(E')$ aus der Menge der Knotenüberdeckungen W des bipartiten Graphen in die Fehlerzustände von G' :

Es sei $\omega = (S \cup T) \in W$ ($S \subseteq X$, $T \subseteq Y$) eine Knotenüberdeckung des bipartiten Graphen G . Dann wird mit

$$Z(\omega) = (V', E \cup E(P) \cup \{(s_d, x) : x \in X - S\} \cup \{(t, y) : y \in Y - T\}) \quad (1.4)$$

ein Ausfallzustand von G' dadurch erzeugt, dass E_X und E_Y nur Kanten enthalten, die zu keinem Knoten aus ω inzident sind (siehe Abbildung 1.4). Es soll gezeigt werden, dass dies eine Bijektion ist:

Sei $\omega = (S \cup T)$ eine Knotenüberdeckung des bipartiten Graphen G . Wir nehmen an, dass $x \in X - S$ und $y \in Y - T$. Damit ist (x, y) keine Kante in $Z(\omega)$, ansonsten wäre sie nicht durch ω abgedeckt. Folglich existieren keine D -Wege in $Z(\omega)$.

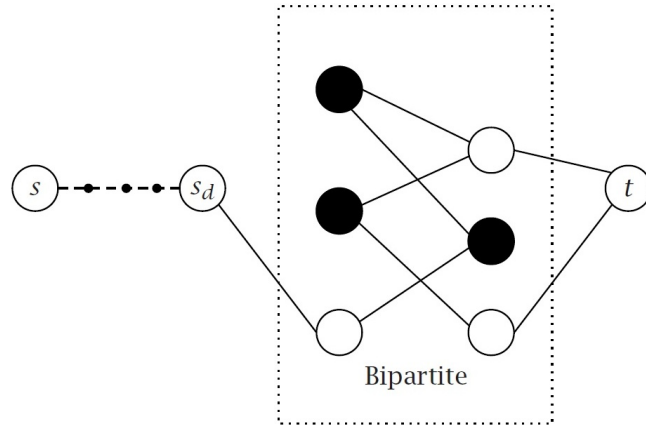


Abbildung 1.4: Ausfallzustand, [Can04]

Zum Verständnis: $D = d + 3$. Zuerst muss P durchlaufen werden, was eine Länge von d benötigt. Danach steht noch eine Länge von 3 zur Verfügung: Eine Kante aus E_X , um X zu erreichen, eine, um von X nach Y zu gelangen und eine aus E_Y , um von Y aus t zu erreichen. Es besteht jedoch folgendes Problem: Die Kanten aus E_X und E_Y gehören zu Knoten aus $X - S$ bzw. $Y - T$, die nicht in der Knotenüberdeckung ω enthalten und deshalb über keine Kante verbunden sind. Folglich kann kein D -Weg existieren. Aus diesem Grunde ist $Z(\omega)$ ein Fehlerzustand von G' .

Zur Eineindeutigkeit von (1.4) fehlt der Nachweis, dass aus einem Fehlerzustand eine Knotenüberdeckung konstruierbar ist. Sei dazu H ein Zustand wie oben beschrieben. Wenn H ein Fehlerzustand ist, müssen die zu s_d und t adjazenten Knoten aus X und Y eine unabhängige Menge bilden (ansonsten wären wenigstens zwei Knoten aus X und Y adjazent und es gäbe einen D -Weg). Daher bilden die verbleibenden Knoten des bipartiten Graphen G eine Knotenüberdeckung. Diese Überlegung entspricht $Z^{-1}(H)$. Wir erinnern uns an die Darstellung des DC2RP über die Fehlerzustände (1.1):

$$R_{s,t}(G, D) = 1 - \sum_{H \in \overline{O}_{s,t}^D(E)} \prod_{e \in H} p_e \prod_{f \notin H} q_f$$

Eingesetzt in die DC2RP-Darstellung der obigen Kantenwahrscheinlichkeiten erhält man:

$$\begin{aligned} R_{s,t}(G', D) &= 1 - \sum_{i=0}^{|X|-|Y|} F_i(G', \{s, t\}, D) \left(\frac{1}{2}\right)^i \left(\frac{1}{2}\right)^{|X|+|Y|-i} (1)^{|E|+|E(P)|} \\ &= 1 - \frac{\sum_{i=0}^{|X|+|Y|} F_i(G', \{s, t\}, D)}{2^{|X|+|Y|}} \end{aligned}$$

Mit Z ist $\sum F_i(G', \{s, t\}, D)$ gleich der Anzahl der Knotenüberdeckungen des bipartiten Graphen G . Damit folgt die Behauptung.

□

Das in dieser Arbeit anzuwendende Verfahren zur Berechnung des Wahrscheinlichkeitspolynoms basiert auf der Verwendung einer Baumzerlegung des Graphen. Das benutzte Prinzip beruht darauf, sich mögliche baumartige Strukturen des Graphen zu Nutze zu machen. Denn viele Probleme auf Graphen werden leichter, wenn der untersuchte Graph ein Baum ist.

Definition 1.3.4 (Baum)

*Ein kreisfreier, zusammenhängender, nicht leerer Graph ist ein **Baum**.*

Beispielsweise ist das Problem der Berechnung des DC2RP NP-schwer für $D \geq 3$. Bei einem Baum hingegen ist die Lösung in polynomialer Zeit berechenbar. Dazu müssen lediglich alle Kanten auf dem einzigen Weg von s nach t intakt sein. Mit der Menge $s, t - P$ der Kanten dieses s, t -Weges lautet das Wahrscheinlichkeitspolynom dementsprechend $p^{|P_{s,t}(D)|}$ (siehe dazu Abbildung 1.5).

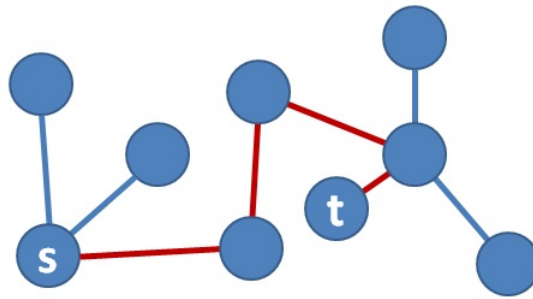


Abbildung 1.5: $R_{s,t}(\text{Baum}, 4, p) = p^4$

Eine Baumzerlegung stellt den Graphen in einer baumartigen Form dar, mit der eine Zuverlässigkeitsberechnung ohne Weglängenbeschränkung in polynomialer Zeit möglich ist ([Gal05], [Luc96], [Luc97]), allerdings nur für Graphen mit beschränkter Baumweite. Es ist folglich von Interesse, ob die Polynomialität auch für Weglängenbeschränkung erhalten bleibt. In dieser Arbeit wird versucht, eine Methode zu finden, welche die Beschränkung D und das Berechnungsverfahren mittels Baumzerlegung vereint.

2 Baumzerlegung

2.1 Einführung

Wie bereits angesprochen, existieren viele NP-schwere Graphenprobleme. Unter anderem auch die Berechnung der Zuverlässigkeit in probabilistischen Graphen, sei es nun für den K -terminalen, all-terminalen oder s, t -terminalen Zusammenhang ([Col87], [Bal80]). Seit den 70-ern des vergangenen Jahrhunderts wird nach Möglichkeiten für eine effiziente Berechnung geforscht. Das Problem der Berechnung ist, wie im letzten Abschnitt gezeigt, NP-schwer und die Laufzeit wächst exponentiell bei größer werdender Anzahl von fehleranfälligen Komponenten wie Kanten oder Knoten. Lediglich bei einigen speziellen Graphenklassen existieren Lösungsalgorithmen mit polynomialer Laufzeit ([Hag83]). Das Problem besteht also für die meisten Klassen von Methoden. Wie wir später in diesem Abschnitt noch sehen werden, kann die Verwendung einer Baumzerlegung hierin eine Ausnahme bilden. Denn Bäume sind eine Graphenklasse, bei denen Berechnungen zur Zusammenhangswahrscheinlichkeit einfach werden können, da es für sie polynomiale Algorithmen gibt.

Ein kurzes Beispiel: Gegeben sei ein Baum $G = (V, E)$. Ist beispielsweise die K -terminale Zuverlässigkeit zu ermitteln, so muss auf Grund der Baumstruktur ein Teilbaum T existieren, der alle K -Knoten beinhaltet und dessen Blätter allesamt K -Knoten sind (siehe Abbildung 2.1). Sei $E(T)$ die Kantenmenge von T . So ist der Zusammenhang nur gewährleistet, wenn alle diese Kanten intakt sind. Das Zusammenhangspolynom lautet demzufolge: $p^{|E(T)|}$.

Der Teilbaum entsteht, indem alle Blätter (Knoten mit Knotengrad 1) gesucht und entfernt werden, die keine K -Knoten sind. Dieser Vorgang wird so lange wiederholt, bis keine Blätter mehr entfernt werden können.

Es ist leicht einzusehen, dass sowohl die Suche nach dem Teilbaum, als auch die Berechnung des Polynoms polynomial in der Größe des Graphen sind.

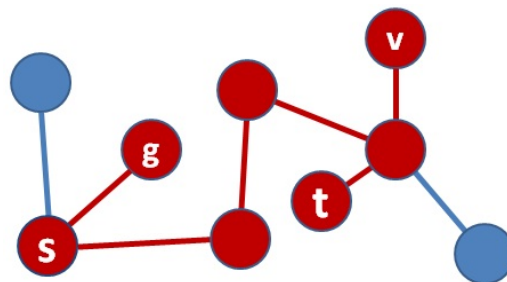


Abbildung 2.1: Baum mit Teilbaum T (rot) und K -Knoten g, s, t, v

Um eine solche Baumzerlegung aufzustellen, werden die von Robertson und Seymour eingeführten Definitionen ([Rob83], [Rob86]) verwendet:

Definition 2.1.1 (Baumzerlegung)

Die **Baumzerlegung** eines Graphen $G = (V, E)$ ist ein Paar (T, ϕ) mit einem Baum $T = (W, F)$ und einer Abbildung $\phi : W \mapsto 2^V$, die jedem Baumknoten $w \in W$ von T eine Graphenknotenteilmenge $X_w \in 2^V$ von G derart zuordnet, dass:

1. $\bigcup_{w \in W} \phi(w) = V$
2. $\{u, v\} \in E \Rightarrow \exists w \in W : \{u, v\} \subseteq \phi(w)$
3. Wenn $v \in W$ auf dem Weg zwischen u und w in T liegt, gilt: $\phi(u) \cap \phi(w) \subseteq \phi(v)$

Wobei $\phi(w) = X_w$ zu setzen ist.

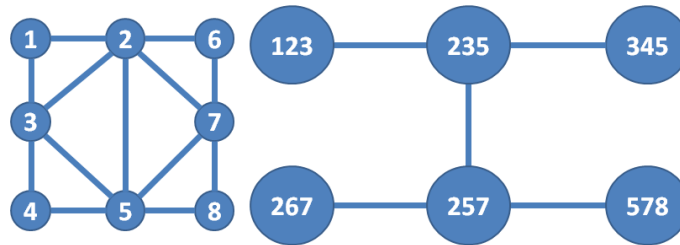


Abbildung 2.4: Beispiel für einen Graphen und seine Baumzerlegung

Die erste Bedingung besagt, dass alle Knoten aus V in den Baumknoten vorhanden sein müssen. Mit der Zweiten wird abgesichert, dass es für jede Kante aus E mindestens einen Baumknoten gibt, der beide Endknoten der Kante enthält. Die Dritte lässt sich folgendermaßen interpretieren: Alle Baumknoten, die den Graphenknoten v enthalten, müssen zusammenhängend sein.

Die **Wegzerlegung** ist ein Spezialfall der Baumzerlegung; sie liegt vor, wenn T ein Weg ist. Des Weiteren werden die üblichen Namen der Baumknoten übernommen:

Baumknoten mit Knotengrad 1 heißen **Blätter**. Verzweigungsknoten nennt man **Join-knoten** (kurz: **Joins**).

Mit der **Weite** wird eine wichtige Indikatorvariable eingeführt:

Definition 2.1.2 (Weite einer Baumzerlegung (bzw. Wegzerlegung))

Die **Weite** einer Baumzerlegung (bzw. Wegzerlegung) ist

$$\max\{|\phi(w)| : w \in W\} - 1$$

Die **Baumweite** (**Wegweite**) $tw(G)$ von G ist die kleinste Weite aller Baumzerlegungen (bzw. Wegzerlegungen) von G .

In dieser Arbeit wird mit **schönen Baumzerlegungen** gearbeitet (siehe Abbildung 2.5). Eine solche liegt vor, wenn sich die Mengen X zweier benachbarter Baumknoten in maximal einem Element unterscheiden und die Blätter die Mächtigkeit 1 besitzen (für diese Arbeit sollen die Blätter nur die leere Menge enthalten) ([Bod98]).

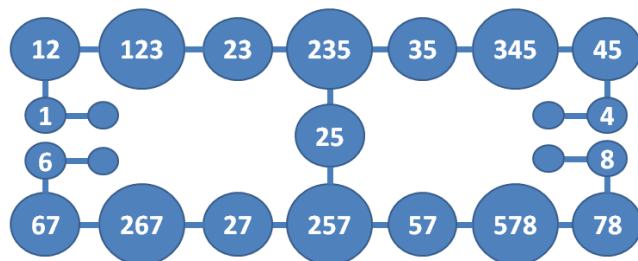


Abbildung 2.5: Erweiterung von Abbildung 2.4 zur schönen Baumzerlegung

2.1.1 Anordnung der Baumzerlegung

Um ein besseres Handling zu ermöglichen, muss zu Beginn der Baum als Wurzelbaum angeordnet werden. Dazu wird ein beliebiges Blatt als Wurzel gewählt und die Entfernung 0 zugewiesen. Anschließend werden die restlichen Knoten ihrem Abstand von der Wurzel entsprechend angeordnet (optische Umsetzung: siehe Abbildung 2.1.1). Betrachtet man sich einen Knoten v im Baum, so heißen alle höherliegenden Knoten auf dem Weg zur Wurzel **Vorgänger** oder **Vorfahren**, die tieferliegenden Knoten mit v als Wurzel **Nachfolger**. Einen benachbarten Vorfahren nennt man **Vater** oder **Elternknoten**, benachbarte Nachfahren **Kind** bzw. **Kinder** (siehe Abbildung 2.6).

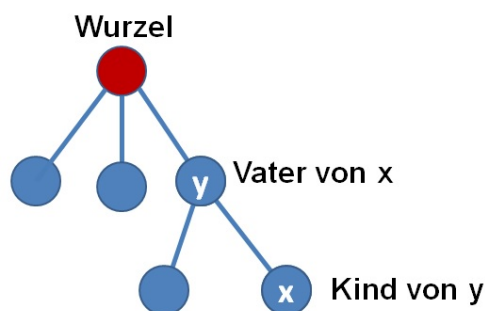


Abbildung 2.6: Darstellung von Wurzel, Vater und Kind

Um ein besseres Umfeld für die Auswertung zu erhalten, kann der Baum der Zerlegung zu einem **Binärbaum** (einem Wurzelbaum dessen Knoten jeweils maximal zwei Kinder besitzen) modifiziert werden (siehe Abbildung 2.7):

Angenommen, die Zerlegung besitzt einen Joinknoten v mit Knotengrad $\deg(v) = x$, $x \in \mathbb{N}$, $x > 3$. Dann dupliziere man v genau $x - 3$ -mal. Die Duplikate werden als Ast an das Original angehängt. Als nächstes wird für jedes im Ast enthaltene Duplikat ein

beliebiger Nachbarknoten von v (außer dem Duplikat) ab- und an ein Duplikat angehängt. Damit erhöht sich der Knotengrad jedes Duplikats von zwei auf drei. Außer bei dem Duplikat, welches das Blatt bildet. Für dieses muss ein zusätzlicher Nachbar von v ab- und an das Blatt angehängt werden. Jedes Duplikat, einschließlich des Astes, besitzt nun Knotengrad drei. v verliert dementsprechend $x - 3 + 1$ Nachbarn, gewinnt jedoch mit dem Ast $+1$ Nachbarn hinzu. Damit ergibt sich für den Knotengrad $\deg(v) = x - (x - 3 + 1) + 1 = 3$. Damit führt die Umverkettung zum gewünschten Ergebnis.

Dieses Vorgehen wird nun für alle Joinknoten $w \in W$ mit $\deg(w) > 3$ durchgeführt. Es ist leicht einzusehen, dass die Umformung eine lineare Laufzeit in Anspruch nimmt, da die Anzahl der Baumknoten den Knotengrad und damit die Anzahl der zu erzeugenden Duplikate beschränkt.

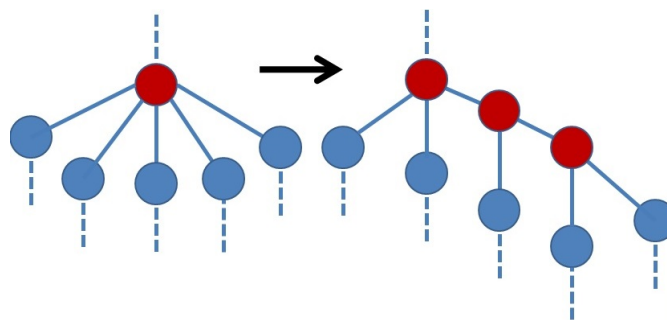


Abbildung 2.7: Entwicklung eines Binärbaumes

2.2 Die Verwendung einer Baumzerlegung

Weiterhin ist zu erläutern, wie die Baumzerlegung bei der Berechnung der Zuverlässigkeit helfen soll. Wie bereits gesagt, bildet eine solche Zerlegung eine andere Art der Darstellung eines Graphen. Die Knoten sind ebenso vollständig enthalten wie die Kanten. Um den Zusammenhang zu berechnen, können die Elemente des Graphen nacheinander untersucht werden.

Gesucht ist ein geeignetes Vorgehen zur Auswertung der Zerlegung. Nachfolgend werden die verschiedenen Methoden beschrieben. Ausgehend vom Standardverfahren für den all-terminalen Zusammenhang werden die Modifikationen für die K -terminale und die s, t -terminale Zuverlässigkeit erläutert.

2.2.1 Beschreibung der Arbeitsweise mit Baumzerlegung

Die Arbeitsweise mit einer Baumzerlegung besteht in der günstigen Anwendung von separierenden Knotenmengen. Diese werden durch die Graphenknotenteilmengen der

Baumknoten dargestellt:

Definition 2.2.1 (Separierende Knotenmenge)

Seien $G^1 = (V^1, E^1)$ und $G^2 = (V^2, E^2)$ Untergraphen von $G = (V, E)$. Erfüllen sie folgende Bedingungen:

$$V^1 \cap V^2 = U \neq \emptyset$$

$$V^1 \cup V^2 = V$$

$$E^1 \cap E^2 = \emptyset$$

$$E^1 \cup E^2 = E$$

dann nennt man die Knotenmenge U **separierende Knotenmenge** von G .

Bemerkung 2.2.1

Durch die Baumzerlegungen erzeugt man mehrere separierende Knotenmengen, die während des unten beschriebenen Voranschreitens in der Zerlegung mittels Hinzufügen oder Entfernen von Knoten verschoben werden. Eine solche sich verschiebende separierende Knotenmenge soll als **Front** bezeichnet sein. Denn die Menge stellt die Trennlinie zwischen bereits bearbeiteten Knoten und unbearbeiteten Knoten, entsprechend zweier Gebiete, dar (siehe Abbildung 2.8).

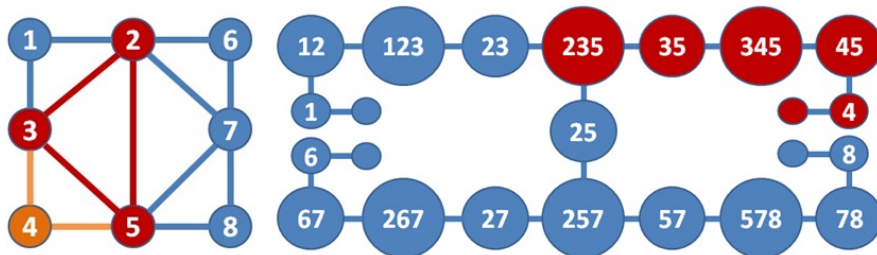


Abbildung 2.8: Orange: bereits deaktiviert; Rot: aktive Menge (Front); Blau: unbearbeitet

Da die Baumzerlegung eine Ordnung der Knoten über die Entfernung zur Wurzel besitzt, kann man das sogenannte **Bottom-Up-Verfahren** verwenden ([Bod97]). Wie der Name bereits sagt, arbeiten wir uns bei diesem Verfahren von unten nach oben durch die Baumzerlegung und gewinnen dabei Informationen (in diesem Fall über den Zusammenhang). Das Durcharbeiten erfolgt, ebenso wie die Informationsweitergabe, Knoten für Knoten, Ast für Ast bis nach oben zur Wurzel. Begonnen wird bei den Blättern. Möchte man aus einem Knoten v der Zerlegung Informationen gewinnen, so werden zuerst alle Informationen des Teilbaums benötigt, dessen Wurzel v bildet. Die Joinknoten nehmen dabei eine bedeutende Stellung ein. Da sie zwei Kinder besitzen, vereinen die Joinknoten die Informationen zweier Teilbäume.

Bemerkung 2.2.2

Der Weg in einer Baumzerlegung zwischen zwei Joinknoten, zwei Blättern oder einem Blatt und einem Joinknoten bildet eine Wegzerlegung für den Untergraphen, der aus den Graphenknoten der betreffenden Baumknoten und allen Kanten zwischen ihnen besteht.

Ein speichergünstiges Verfahren für Bottom-Up liefert dabei folgendes Vorgehen:

Algorithm 1 Bottom-Up-Algorithmus

bottom-up(Baumzerlegung, Wurzel):

if Wegzerlegung liegt vor **then**

 Beginne bei dem einzigen Blatt neben der Wurzel und bearbeite den Weg vom Blatt zur Wurzel

return Wurzelinformation

else

 Rücke vor zum nächsttieferen Join A

 Bilde Informationen des linken Teilbaums: bottom-up(linker Teilbaum, Join A)

 Bilde Informationen des rechten Teilbaums: bottom-up(rechter Teilbaum, Join A)

 Vereinige die Informationen der beiden Teilbäume im Join A

 Bearbeite den Weg von Join A zur Wurzel

return Wurzelinformation

end if

Durch diesen Algorithmus entspricht die Anzahl der zu speichernden Informationen der Tiefe des tiefsten Blattes im Baum + 1; denn in allen Joinknoten entlang des Weges vom Blatt zur Wurzel (und der Wurzel selbst) können schon Informationen genau eines Teilbaums sein. Das Blatt enthält dann eine weitere weiterzugebende Information. Es bleibt zu klären, wie die Informationen erzeugt und gespeichert werden.

Wichtig ist hierbei der Vergleich zweier benachbarter Baumknoten („Kind“ und „Vater“). Genauer gesagt, werden ihre Graphenknotenteilmengen X_{Kind} und X_{Vater} verglichen. Dabei können auf Grund der vorausgesetzten schönen Baumzerlegung nur drei mögliche Ereignisse eintreten:

X_{Vater} enthält einen Graphenknoten mehr/weniger als X_{Kind} oder $X_{\text{Vater}} = X_{\text{Kind}}$.

Im letzteren Fall wird die Information lediglich vom Kind auf den Vater übertragen. In den anderen beiden Fällen aber wird die Front der separierenden Knotenmenge durch die Erweiterung/Entfernung von Knoten verschoben. Jeder Ast beginnt dabei an einer anderen Stelle des Graphen und die dabei untersuchten Knoten nebst Kanten bilden Teilgraphen. In dieser Arbeit ist ein **Teilgraph** ein Untergraph von G' , der aus den Knoten und Kanten besteht, die während des Durchlaufens der Baumzerlegung betrachtet

werden. Im Joinknoten überlagern sich zwei derartige Graphen G'_1 und G'_2 , so dass die Fronten verschmelzen und aus zwei Teilgraphen einer wird.

Mit jedem neuen Knoten wird der entsprechende Teilgraph derart erweitert, dass die im Originalgraphen existierenden Kanten zwischen alter Front und neuem Knoten auch im Teilgraphen vorhanden sind.

2.2.2 Partitionen als Informationsspeicher

Noch nicht geklärt wurde, wie man das Zuverlässigkeitspolynom erzeugt. Da ein probabilistischer Graph mit der Baumzerlegung dargestellt wird, bedarf es einer Speicherform, die Aussagen über die verschiedenen Zustände der Kanten, des Zusammenhangs des Graphen oder der Teilgraphen und der jeweiligen Wahrscheinlichkeit liefert.

Eine solche Form lieferten beispielsweise Lucet, u.a. in [Luc96], [Luc97] mit der Idee, den bis zur aktuellen Front erzeugten Zusammenhang in Partitionen festzuhalten.

Konkret besagt die Idee Folgendes: Gespeichert werden Aussagen zur Front. Durch Ausfall oder Intaktheit jeder neuen Kante können alle möglichen Zustände eines Teilgraphen generiert werden. Hierbei ist nur der Zusammenhang der aktuellen Front von Interesse. Darum können alle diejenigen Zustände nebst ihren Eintrittswahrscheinlichkeiten zusammengefasst werden, die zum Zusammenhang derselben Frontknoten führen. Zur Darstellung dieses Zusammenhangs dienen Partitionen, die aus der Menge aller Frontknoten gebildet werden.

Bei einem Zustand des Teilgraphen liegen zwei Frontknoten dann und nur dann in einem Block, wenn sie durch einen Weg aus intakten Kanten verbunden sind. Die Wahrscheinlichkeiten aller Zustände, die zu ein und derselben Partition führen, werden addiert und die Zustände somit zusammengefasst. Dabei ist zu beachten, dass ein Teilgraph mehrere unterschiedliche Zustände und damit mehrere unterschiedliche Partitionen besitzen kann. Die maximale Anzahl B_n von Partitionen einer n -elementigen Menge ist die Bellzahl (siehe Kapitel 2.5).

2.2.3 Formulierung der vier Grundschritte

Wie oben bereits beschrieben, werden zur Untersuchung der Baumzerlegung vier relevante Grundschritte angewandt:

1. *Hinzufügen eines Knotens (Aktivierung)*
2. *Entfernen eines Knotens (Deaktivierung)*
3. *Untersuchung einer Kante (Kantenuntersuchung)*
4. *Zusammenführung der Informationen zweier Äste (Fusion)*

Zur Beschreibung der vier Grundschritte wollen wir die in [Pön01] verwendeten Formeln und Begriffe nutzen:

Für eine Teilmenge $X \subseteq V$ sei $\mathbb{P}(X)$ die **Menge aller Partitionen von X** . Sei $X_i \subseteq$ die Front in Schritt i . Jeder neue Schritt in der Baumzerlegung erzeugt eine Menge von Zuständen. In diesem Kontext ist ein **Zustand** ein Paar (π, P_π) , wobei $\pi \in \mathbb{P}(X_i)$ und $P_\pi = P(G^i, \pi)$. Hierbei beschreibt G^i den bis zum Schritt i erzeugten Teilgraphen. Sei Z_i die **Menge aller Zustände in Schritt i** . $(\{\emptyset\}, 1)$ mit leerer Partition und Polynom 1 wird als Startzustand in allen Blättern (außer der Wurzel) initialisiert.

Sei π eine Partition von X . Man schreibt $Y \in \pi$, wenn Y ein Block von π ist. Sei $M(\pi, v)$ die Menge aller Partitionen, die aus π hervorgehen, wenn das Element v in die verschiedenen Blöcke der Partition integriert wird. Die Partition $\pi|\{v\}$ entsteht, indem v als Einzelblock in π aufgenommen wird. Eine Kante $e = \{u, v\}$ kann als Partition angesehen werden, die u und v in einem Block enthält und Einzelblöcke entsprechen allen anderen Knoten. Dann entsteht das **Supremum** $\pi \vee e$ als Partition aus π , indem die Blöcke, in denen die Endknoten von e liegen, vereinigt werden.

Die 4 möglichen Grundschritte im Schritt i erzeugen eine Transformation $\chi : 2^{Z_i} \rightarrow 2^{Z_{i+1}}$ der Zustandsmenge. Sei dazu Abbildung 2.9 mit zwei Beispielszuständen gegeben:

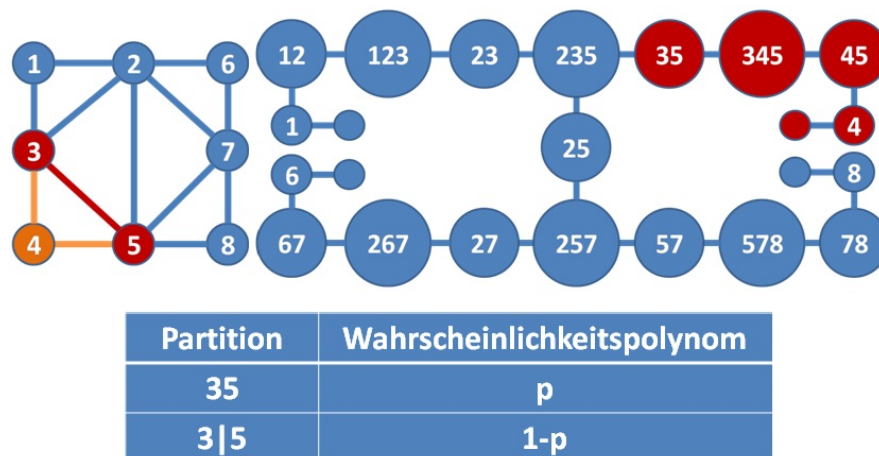


Abbildung 2.9: Ausgangsbeispiel für nachfolgende Grundschritte mit Beispielpolynom

Aktivierung des Knotens v

Es gelte für die Graphenknotenteilmengen zweier benachbarter Baumknoten $|X_i - X_{i-1}| = 1$. Wird nun X_i zur Front, so entsteht ein Teilgraph G^i aus G^{i-1} durch Aufnahme (Aktivierung) von v in den Teilgraphen G^{i-1} . Bis zum Zeitpunkt der Aktivierung kann es somit keine v als Endknoten enthaltenden Kanten geben. Demzufolge bildet v als einzelner Knoten eine Komponente. In den Partitionen ist er als Einzelblock einzufügen. Kanten werden dabei nicht betrachtet, also hat die Aktivierung von v keinen

Einfluss auf die Wahrscheinlichkeitspolynome der Zustände.

$$\{(\pi, P_\pi)\} \mapsto \{(\pi|_{\{v\}}, P_\pi)\}$$

Abbildung 2.10 stellt hierbei den Schritt dar, der auf Abbildung 2.9 folgt.

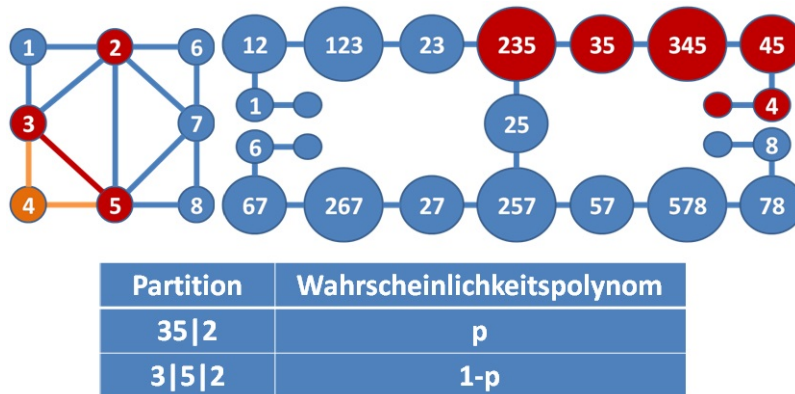


Abbildung 2.10: Aktivierung eines Knotens

Deaktivierung des Knotens v

Es gelte für die Graphenknotenteilmengen zweier benachbarter Baumknoten

$$|X_{i-1} - X_i| = 1$$

Wird X_i zur Front, so entsteht sie aus der alten Front von X_{i-1} durch Entfernung (Deaktivierung) des Knotens v . Die Abbildungen 2.11 und 2.12 stellen hierbei den Schritt nach Abbildung 2.10 und der Kantenuntersuchung (siehe Abbildungen 2.13 und 2.14) dar.

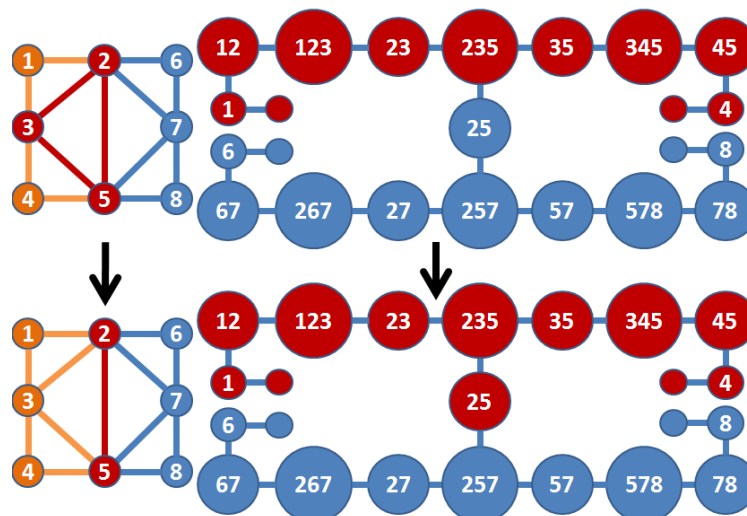


Abbildung 2.11: Umsetzung der Deaktivierung im Graph

Dabei können in den Partitionen der Zustände drei verschiedene Situationen eintreten:

1. Wenn $\{v\}$ ein Einzelblock von π ist und $|\pi| > 1$, dann gilt:

$$(\pi, P_\pi) \mapsto \emptyset$$

Der Block mit v als Einzelknoten wird aus der aktiven Menge entfernt und diese ist danach nicht leer. Dies bedeutet, dass v und alle vorher in diesem Block befindlichen (und daher mit v verbundenen) Knoten nun eine Komponente bilden. Da $|\pi| > 1$, gibt es weitere Frontknoten, die nicht mit den Knoten des v -Blocks verbunden sind. Durch die Deaktivierung ist der Block nicht mehr untersuchbar. Damit kann im weiteren Verlauf der Untersuchung weder v noch einer der anderen Blockknoten in der unten erklärten Kantenuntersuchung mehr auftreten. Folglich gibt es keine Möglichkeit, dass eine Kante eingefügt und die Blockknoten mit anderen Knoten verbunden werden. Eine Komponente ist entstanden und der Zusammenhang somit nicht mehr erreichbar. Demzufolge sind alle derartigen Zustände zu entfernen.

2. Wenn $\pi = \{v\}$, dann führt die Entfernung dieses letzten Knotens zur leeren Partition

$$(\{v\}, P_{\{v\}}) \mapsto (\emptyset, P_{\{v\}})$$

Da von einem Graphen ohne mehreren Komponenten ausgegangen wird, kann es auf Grund der dritten Bedingung der Definition einer Baumzerlegung keine leeren Baumknoten außer den Blättern geben. In dieser Situation erreicht man also eines der Blätter. Durch die Anordnung der Baumzerlegung gibt es jedoch kein Blatt neben der Wurzel, das ein Vater ist. Da es aber einen zu deaktivierenden Knoten v gibt und die leere aktive Menge auf ein Blatt schließen lässt, muss es sich dabei um die Wurzel handeln.

Damit wurde der gesamte Graph untersucht. Da wegen Punkt 1 alle Zustände mit Komponentenbildung entfernt werden, mussten im letzten entfernten Block alle Graphenknoten verbunden und dadurch enthalten sein. Auf Grund der Tatsache, dass im vorherigen Schritt nur noch v als Einzelblock vorhanden war und dass Zustände mit identischen Partitionen wie unten beschrieben zusammengefasst werden, enthält die Zustandsmenge in diesem letzten Schritt nur einen Zustand. Damit ist das Wahrscheinlichkeitspolynom $P_{\{v\}}$ dieses Zustands das Zuverlässigkeitspolynom des Graphen G .

3. Ansonsten gilt für die restlichen Zustände:

$$\bigcup_{\pi \in M(\sigma, v)} \{(\pi, P_\pi)\} \mapsto \left\{ \left(\sigma, \sum_{\pi \in M(\sigma, v)} P_\pi \right) \right\}$$

Durch die Punkte 1 und 2 gibt es nur noch Zustände mit Partitionen π zu betrachten, bei denen v zusammen mit anderen Knoten in einem Block liegt. Obige Formel besagt, dass alle Zustände mit Partitionen π , bei denen die Entfernung von v zur selben identischen Partition σ führen würde, als Zustand mit Partition σ zusammengefasst werden, indem man ihre Wahrscheinlichkeitspolynome addiert. Partitionen, die in $M(\sigma, v)$, aber nicht in den Zuständen von Z_i enthalten sind, existieren nicht. Ihre Wahrscheinlichkeitspolynome wären mit 0 anzusetzen, wodurch sie sich in der Summe nicht auswirken. Da bei der Deaktivierung keine Kanten untersucht werden, ändern sich die restlichen Wahrscheinlichkeitspolynome nicht. Damit ist ersichtlich, dass jeder Zustand für mehrere Teilgraphen steht, bei denen die Verbindung++++ der Frontknoten zur selben Partition führt. Damit sind alle entstandenen Zustände mit identischen Partitionen zusammengefasst.

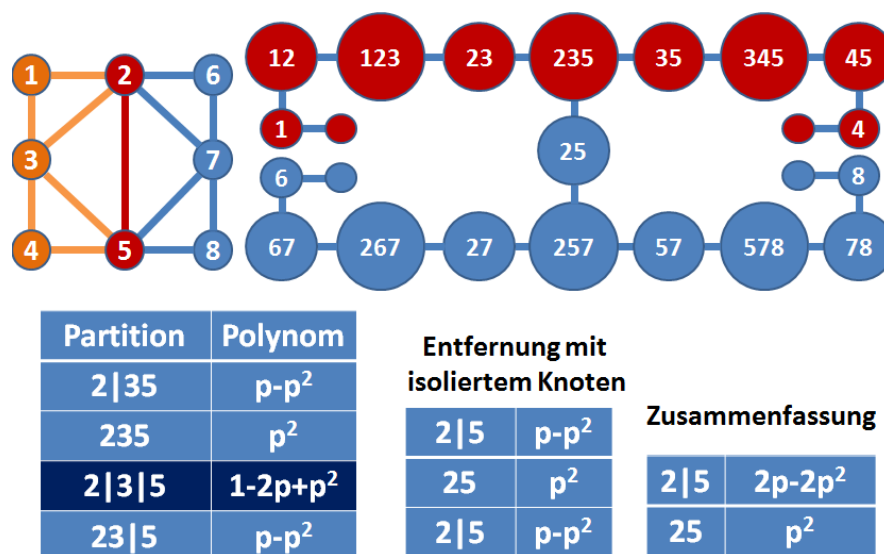


Abbildung 2.12: Umsetzung der Deaktivierung in den Zuständen

Bemerkung 2.2.3

Die Deaktivierung wird auf Grund der Anordnung und der dritten Bedingung der Definition einer Baumzerlegung für jeden Knoten nur einmal durchgeführt.

Beweis(indirekt):

Angenommen v wird zweimal deaktiviert.

Dann gibt es zwei Baumknoten a und a' . Deren Kinder enthalten v bis in eine bestimmte Tiefe. Auf Grund der Anordnung der Zerlegung müssen sich die Eltern von a und a' irgendwann treffen. Die v enthaltenden Kinder von a und a' können sich hingegen nie treffen. Wäre es so, entstünde entweder ein Kreis, oder a wäre o.B.d.A. ein v enthaltendes Kind von a' und v würde im Schritt nach a nicht deaktiviert.

Es entstehen also zwei v enthaltende Teilbäume. Damit befindet sich auf dem Weg zwischen den Teilbäumen mindestens ein v nicht enthaltender Baumknoten. Mit Bedingung 3 der Definition einer Baumzerlegung folgt der Widerspruch. \square

Untersuchung der Kante $e = \{u, v\}$

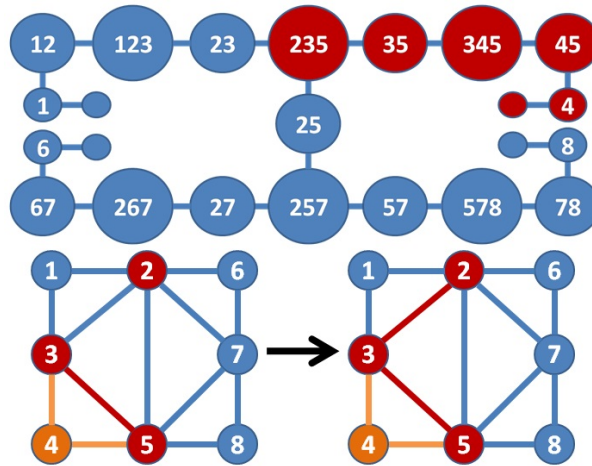


Abbildung 2.13: Umsetzung der Untersuchung im Graph für die Kante $\{2, 3\}$

Die Kantenuntersuchung erfolgt direkt nach der Aktivierung des Knotens v . Untersucht werden dabei diejenigen von v ausgehenden Kanten, deren zweiter Endknoten in der Front liegt. Zusätzlich wird gefordert, dass jede Kante nur einmal während der gesamten Bearbeitung der Baumzerlegung untersucht wird. Angenommen, wir untersuchten eine Kante zweimal, so könnte sie einmal intakt und einmal ausgefallen sein. Dies führt zu einem Widerspruch. Außerdem ginge unter Umständen ihre Wahrscheinlichkeit mehrfach ein.

Für die Untersuchung werden die zwei möglichen Zustände von e generiert: intakt und ausgefallen. Dadurch werden aus jedem Teilgraphen zwei Teilgraphen. Der erste enthält dabei die ausgefallene und der zweite die intakte Kante. Angewandt auf die Zustände bedeutet dies, dass jeder Zustand dupliziert wird und man ihn einmal auf den Ausfall und einmal auf die Intaktheit hin untersucht (siehe dazu in Abbildung 2.14 den Schritt von der linken Tabelle in die mittlere):

Ausfall der Kante Das Polynom des Zustands wird mit der Wahrscheinlichkeit für den Ausfall $(1 - p)$ multipliziert. Da mit dem Ausfall kein Teilgraph verändert wird, muss auch an der Partition keine Veränderung vorgenommen werden. Wir erhalten somit den Zustand

$$(\pi, (1 - p) P_\pi)$$

(Vergleiche dazu in Abbildung 2.14 die mittlere und rechte Tabelle, Zeile 2 und 4)

Intaktheit der Kante Das Polynom des Zustands wird mit der Wahrscheinlichkeit für die Intaktheit p multipliziert. Durch die Intaktheit sind u und v verbunden. Dieser Fakt ist in der Partition erfasst, indem das Supremum $\pi \vee e$ gebildet wird. Man vergleiche dazu

in Abbildung 2.14 den Übergang von der mittleren zur rechten Tabelle, Zeilen 3 und 5. Es ergibt sich folgende Formel:

$$\{(\pi, P_\pi)\} \mapsto \{(\pi, (1-p)P_\pi), (\pi \vee e, pP_\pi)\}$$

Danach werden, wie bei der Deaktivierung, Zustände mit identischen Partitionen zusammengefasst, indem man ihre Wahrscheinlichkeitspolynome addiert.

Während der Untersuchung kann in einem Zustand einer der folgenden Fälle auftreten:

1. *u und v liegen in verschiedenen Blöcken:*

Dann unterscheiden sich die Partitionen der duplizierten Zustände und die Formel kann angewandt werden.

2. *u und v liegen im selben Block:*

Dann unterscheiden sich die Partitionen der duplizierten Zustände nicht, denn $\pi \vee e = \pi$. In diesem Fall würden die duplizierten Zustände am Ende zusammengefasst. Es ergibt sich

$$(\pi, pP_\pi + (1-p)P_\pi) = (\pi, P_\pi)$$

Partition	Polynom	Partition	Polynom
2 35	p	2 35	(1-p)p
2 3 5	1-p	2 35	pp
		2 3 5	(1-p)(1-p)
		2 3 5	p(1-p)

Abbildung 2.14: Umsetzung der Untersuchung der Kante $\{2,3\}$ in den Zuständen

Bemerkung 2.2.4

Ein Zustand von Fall 2 ändert sich nicht. Folglich kann man bei der Kantenuntersuchung von e alle Zustände ignorieren, die dieser Variante angehören und bearbeitet nur die Zustände des ersten Falles. Damit wird Rechenzeit gespart.

Fusion der Teilbäume t_1 und t_2 im Joinknoten v

Eine Fusion tritt immer auf, wenn zwei Teilbäume t_1, t_2 in einem Joinknoten v zusammentreffen (siehe Abbildung 2.15, roter und grüner Teilbaum). Dann werden die Informationen der Teilbäume im Joinknoten zusammengefasst.

Theoretisch geschieht dies, indem je ein Teilgraph $G^1 = (V^1, E^1)$ aus t_1 und ein Teilgraph $G^2 = (V^2, E^2)$ aus t_2 zu einem Teilgraphen $G^3 = (V^3, E^3)$ mit $V^3 = V^1 \cup V^2$ und $E^3 = E^1 \cup E^2$ kombiniert werden. Auf Grund der Einmaligkeit der Untersuchung einer Kante gilt dabei: $E^1 \cap E^2 = \emptyset$. Außerdem ist $V^1 \cap V^2 = X_v$.

Wir haben Zustände eingeführt, um Teilgraphen zusammenzufassen, die zum selben Zusammenhang der Frontknoten führen. Die praktische Umsetzung des obigen Absatzes lautet damit wie folgt:

Angenommen, man vertauscht den Teilgraphen G^1 mit $G^4 = (V^4, E^4)$, der aus demselben Zustand stammt. Dann hängen in G^4 dieselben Frontknoten zusammen wie in G^1 . Lediglich die Wege zwischen den Knoten könnten andere sein. Es genügt also, alle Zustände von t_1 mit allen Zuständen von t_2 zu kombinieren. Dabei wird aus je zwei kombinierten Zuständen ein neuer Zustand gebildet.

Die Kombination der Zustände erfolgt in drei Schritten (siehe Tabelle in Abbildung 2.15):

1. Bilde die Partition des neuen Zustands: \rightarrow Supremum der Partitionen zweier kombinierter Zustände
2. Bilde das Wahrscheinlichkeitspolynom des neuen Zustands: \rightarrow Produkt der Wahrscheinlichkeitspolynome zweier kombinierter Zustände
3. Fasse Zustände mit identischen Partitionen zusammen, indem ihre Wahrscheinlichkeitspolynome addiert werden

Für einen Zustand a aus der Zustandsmenge von t_1 und einen Zustand b aus der Zustandsmenge von t_2 gilt:

$$(\pi_a, P_{\pi_a}) \text{ fusioniert mit } (\pi_b, P_{\pi_b}) \mapsto (\pi_a \vee \pi_b, P_{\pi_a} P_{\pi_b}) \quad (2.1)$$

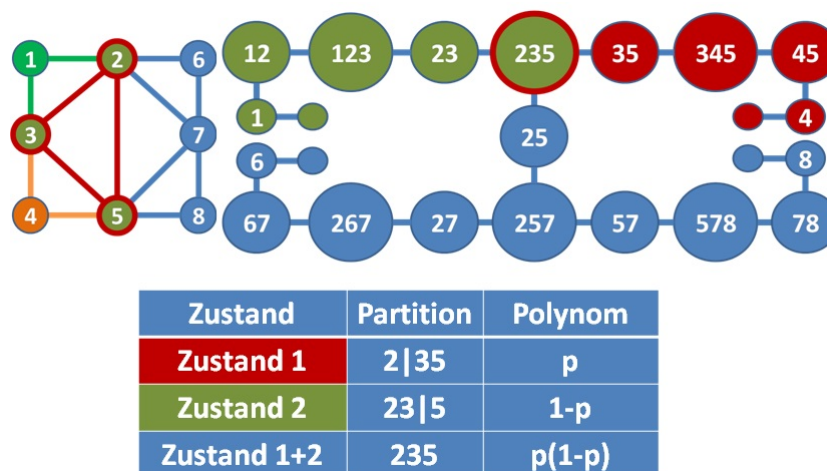


Abbildung 2.15: Umsetzung der Fusion des roten und des grünen Teilbaums

Komplexität der Berechnung des all-terminalen Zusammenhangspolynoms

Satz 2.2.1 (Nach Galtier, Laugier, Pons)

Die Komplexität zur Berechnung der Zusammenhangswahrscheinlichkeit für den vorgestellten Algorithmus liegt in

$$O\left(|V|f(w)^2 + |E|f(w)\right)$$

mit

$$f(w) = \left(\frac{w}{\ln w}\right)^w e^{-(1+o(1))w}$$

und w als Weite der Baumzerlegung.

Zum Beweis dieses Satzes sei auf Kapitel 2.5 verwiesen. Einzig die Anzahl der Elemente einer Partition liegt nicht bei $\max_{i \in I} |X_i \cup \{s, t\}| \leq w + 3$ Elementen, sondern $\max_{i \in I} |X_i| \leq w + 1$ Elementen für den worst case, da noch keine ausgezeichneten Knoten s und t betrachtet werden.

2.3 Erweiterung zur Berechnung des K -terminalen Zusammenhangspolynoms

Für die Methode zur Berechnung des Zusammenhangspolynoms eines probabilistischen Graphen G unter Verwendung der Baumzerlegung soll in diesem Abschnitt eine von [Luc96] vorgestellte Modifikation erläutert werden. Mit dieser ist es möglich, das K -terminale Zusammenhangspolynom $R_K(G, p)$ zu berechnen. Dabei sei $K \subseteq V$. Die Bedingung an den Zusammenhang der Knoten ist damit derart gelockert, dass nun auch Untergraphen mit mehreren Komponenten als zulässig gelten. Allerdings nur solche, bei denen alle K -Knoten in derselben Komponente liegen.

Um wesentliche (K -Knoten enthaltende) von unwesentlichen (K -Knoten nicht enthaltende) Komponenten in den Partitionen zu unterscheiden, sollten nach [Luc96] alle K -Knoten enthaltenden Blöcke markiert werden. Dafür existieren in der Literatur verschiedene Varianten. Für diesen Abschnitt wird vereinfachend angenommen, dass eine Markierung in Form einer Unterstreichung der entsprechenden Blöcke vorgenommen worden ist. Die Auswirkungen äußern sich in den vier Grundschrinen wie folgt: (es werden dabei lediglich die Änderungen im Vergleich zu den oben beschriebenen Grundschrinen aufgeführt)

1. Aktivierung des Knotens v :

Ist $v \in K$, so markiere den Einzelblock $\{v\}$.

2. Deaktivierung des Knotens v :**(a) $\{v\}$ ist ein unmarkierter Einzelblock**

Damit bildet sich eine Komponente ohne K -Knoten; ihr Zusammenhang ist damit noch möglich. Der Zustand wird nicht entfernt.

(b) $\{v\}$ ist ein markierter Einzelblock

Damit bildet sich eine K -Knoten enthaltende Komponente. Wenn alle K -Knoten darin enthalten sind, ist der K -terminale Zusammenhang gegeben. Im anderen Fall werden die Knoten dieser Komponente nicht mehr untersucht, womit der gesuchte Zusammenhang nicht mehr erzeugt werden kann.

3. Untersuchung der Kante $e = \{u, v\}$:

Wird das Supremum zweier Blöcke gebildet, von denen mindestens einer markiert ist, so ist das Supremum zu markieren.

4. Fusion der Teilbäume t_1 und t_2 im Joinknoten v :

Ist mindestens einer der Blöcke einer Supremumsbildung markiert, so markiere das Supremum.

Vorzeitige Entfernung von Zuständen

Mit der Zulassung von Komponentenbildung ist es möglich, dass ein Zustand $T = (\pi_T, P_{\pi_T})$ schon vor Prozessende den K -terminalen Zusammenhang erreicht. Ein solcher Zustand kann vorzeitig entfernt werden. Sein Wahrscheinlichkeitspolynom wird am Ende des Prozesses zum Zuverlässigkeitspolynom addiert. Folgende zwei Bedingungen werden an einen derartigen Zustand gestellt:

(a) Alle K -Knoten befinden sich in einem Block.

Dazu müssen sie keine Frontknoten sein. Es ist notwendig und hinreichend, wenn sich unter den noch zu aktivierenden Knoten keine K -Knoten mehr befinden und π_T genau einen markierten Block enthält. Damit ist der Zusammenhang der K -Knoten gewährleistet.

(b) Punkt a ist erfüllt und es sind keine weiteren Fusionen auszuführen.

Nach der Erfüllung von Punkt a kann sich der Zusammenhang der K -Knoten nicht mehr ändern, wohl aber die K -terminale Zusammenhangswahrscheinlichkeit. Diese ändert sich nach der letzten Fusion nicht mehr, wodurch weitere Berechnungen mit diesem Zustand unnötig werden.

Beweis:

Schritt 1:

Angenommen, es ist keine Fusion mehr auszuführen. Dann wirken sich die verbleibenden ausführbaren Grundschrte auf P_{π_T} derart aus, dass das Polynom mit p und $1 - p$ multipliziert und/oder im Zuge der Deaktivierung zu anderen Polynomen addiert wird.

Lassen wir P_{π_T} bei der Addition mit anderen Polynomen (auf Grund der Zusammenfassung bei Deaktivierung) als einzelnen Summanden bestehen, z.B.: $P_{\pi_T} + P_{\pi_{\text{andere Zustände}}}$, so können die Multiplikationen mit p und $1 - p$ während der weiteren Kantenuntersuchungen auf P_{π_T} und den Restsummanden aufgeteilt werden, z.B.: $pP_{\pi_T} + pP_{\pi_{\text{andere Zustände}}}$.

Am Ende des Prozesses werden alle übriggebliebenen Polynome addiert. Es ist folglich möglich, die Summe folgendermaßen darzustellen:

$$R_K(G, p) = \sum(P_{\pi_T} \text{ enthaltende Polynome}) + \sum(P_{\pi_T} \text{ nicht enthaltende Polynome}) \quad (2.2)$$

Da Punkt a erfüllt ist, wird kein P_{π_T} -enthaltender Zustand gelöscht. Nach oben beschriebener Auswirkung der Kantenuntersuchung lässt sich damit $\sum(P_{\pi_T} \text{ enthaltende Polynome})$ durch geschicktes Ausklammern umschreiben:

$$\sum(P_{\pi_T} \text{ enthaltende Polynome}) = (p + 1 - p)(p + 1 - p) \dots (p + 1 - p) P_{\pi_T} = P_{\pi_T} \quad (2.3)$$

Setzt man nun (2.3) in (2.2) ein, so erhält man:

$$R_K(G, p) = P_{\pi_T} + \sum(P_{\pi_T} \text{ nicht enthaltende Polynome})$$

was der Beschreibung bei vorzeitigem Entfernen von T entspricht.

Schritt 2:

Angenommen, es ist noch genau eine Fusion auszuführen. O.B.d.A. sei angenommen, dass Teilbaum t_1 lediglich den Zustand T_1 und Teilbaum t_2 genau einen Zustand $T_2 = (\pi_{T_2}, P_{\pi_{T_2}})$ enthält. Durch die Fusion entsteht der Zustand

$$T_3 = (\pi_{T_1} \vee \pi_{T_2}, P_{\pi_{T_1}} P_{\pi_{T_2}})$$

Da T_3 mit $\pi_{T_1} \vee \pi_{T_2}$ Punkt a erfüllt und die letzte Fusion gerade abgeschlossen wurde, gilt:

$$R_K(G, p) = P_{\pi_{T_1}} P_{\pi_{T_2}} \quad (2.4)$$

Sei T_1 vor der Fusion vorzeitig entfernt worden. Dann ist der Grundschrte nicht auszuführen. T_2 wird damit bis zum Ende des Prozesses an Stelle von T_3 bearbeitet. Der Einfachheit wegen soll T_2 auch Punkt a erfüllen. Für die Zuverlässigkeit ergibt sich:

$$R_K(G, p) = P_{\pi_{T_1}} + P_{\pi_{T_2}} \quad (2.5)$$

Es ist also $(2.4) \neq (2.5)$.

Da T_1 , bzw. $P_{\pi_{T_1}}$ bei Fusion mit anderen Polynomen multipliziert wird, ist prinzipiell nicht davon auszugehen, dass eine Summendarstellung wie bei (2.2) erreichbar ist. Für die Fusion mehrerer Zustände ist bei Anwendung der vorzeitigen Entfernung von Zuständen also ein ähnlicher Fehler anzunehmen, wenn Entfernungen vor der letzten Fusion erfolgen. \square

2.3.1 Komplexität

Im worst case erfolgt die letzte Kantenuntersuchung vor den letzten beiden Deaktivierungen. Der Unterschied zur Komplexität der all-terminalen Zuverlässigkeit ist damit konstant in der Anzahl der Deaktivierungen. Die Komplexität ändert sich somit nur geringfügig.

Es lassen sich verschiedene Kriterien zur Wahl der Wurzel treffen. Sie kann beispielsweise so gewählt werden, dass die letzte Fusion und die letzte Kantenuntersuchung möglichst früh eintreten, um viele Zustände entfernen zu können.

2.4 Erweiterung zur Berechnung des s, t -Zusammenhangs

Als Spezialfall der Berechnung des K -terminalen Zusammenhangspolynoms ist auch hier für die s und/oder t enthaltenden Blöcke eine Markierung zu verwenden. Sie erfolgt durch die Knoten s und t selbst. Blöcke sind genau dann markiert, wenn sie s und/oder t enthalten.

Die Idee dahinter ist, dass s und t auch nach ihrer Deaktivierung in den Blöcken und der aktiven Menge als imaginäre Knoten erhalten bleiben. Dafür sind die Knoten mit s' und t' zu kennzeichnen. Sie werden also bei der Bildung eines Supremums wie die anderen Knoten behandelt. Eine Ausnahme bildet dabei die Deaktivierung: Ein Block gilt genau dann als leer, wenn er leer ist oder s' , t' oder s' und t' und keine weiteren Knoten enthält. Entsprechend der Markierung werden bei der Deaktivierung unterschiedliche Vorgehensweisen gewählt.

2.5 Komplexität zur Berechnung des s, t -Zusammenhangspolynoms

Im Vergleich zur Berechnung des K -terminalen Zusammenhangspolynoms ergeben sich nach Aussage des obigen Abschnittes keine Unterschiede. Die Komplexitäten des $R(G, p)$, des $R_K(G, p)$ und des $R_{s,t}(G, p)$ sind also identisch. Um die Komplexität eines solchen Algorithmus anzugeben, beziehen wir uns auf die Version von [Gal05]. Diese

Variante der Berechnung unterscheidet sich von der hier vorgestellten im vorzeitigen Entfernen von Zuständen und der Markierung der s, t -Blöcke:

Satz 2.5.1 (Galtier, Laugier, Pons)

Die Komplexität zur Berechnung der s, t -Zusammenhangswahrscheinlichkeit für den vorgestellten Algorithmus liegt in

$$O\left(|V|f(w)^2 + |E|f(w)\right)$$

mit

$$f(w) = \left(\frac{w}{\ln w}\right)^w e^{-(1+o(1))w}$$

und w als Weite der Baumzerlegung.

Beweis:

Die Komplexität des Algorithmus kann mittels der Verwendung der vier Grundschriffe abgeschätzt werden. Sei I die Indexmenge der Baumknoten. Wir haben vorausgesetzt, dass sich zwei adjazente Baumknoten X_i, X_j mit $i \neq j$ maximal um je einen Graphenknoten unterscheiden. Damit gibt es maximal $2|V|$ Elemente im Baum T und deshalb $|V|$ Knoten mit zwei oder mehr Kindern.

Außerdem enthält jede Partition maximal $\max_{i \in I} |X_i \cup \{s, t\}| \leq w + 3$ Elemente.

Eine Fusion von k Kindern kann in $(k-1)B^2(w+3)O(w \log^*(w))$ Operationsschritten erreicht werden. Dabei stellt $O(w \log^*(w))$ die Komplexität zur Erzeugung eines Supremums zweier Partitionen dar und die Bellzahl $B(w+3)$ die Anzahl der erstellbaren Partitionen einer $w+3$ -elementigen Menge. Dies führt bei einer Komplexität von $O(|V|B^2(w+3)w \log^*(w))$ zur Durchführung aller Fusionen.

Die Komplexität der Untersuchung einer Kante liegt in $O(B(w+3))$.

Während des Durchlaufes durch den Baum wird jeder Knoten nur einmal entfernt. Dies benötigt eine Zeit von $O(B(w+3))$ (analog zur Aktivierung).

Darum liegt die Anzahl der benötigten Schritte des Algorithmus in $O(|V|B^2(w+3)w \log^*(w) + |E|B(w+3))$

Verwendet man die obige Formel, erhalten wir $O(|V|f(w)^2 + |E|f(w))$ mit $f(w) = \left(\frac{w}{\ln w}\right)^w e^{-(1+o(1))w}$

Letztere Formel beschreibt die Bellzahl nach der asymptotischen Formel von de Bruijn [Bru58]

$$\frac{\ln(B(n))}{n} = \ln(n) - \ln \ln(n) - 1 + o(1)$$

□

3 Berechnung der weglängenbeschränkten Zuverlässigkeit

Nachdem in den vorangegangenen Kapiteln alle Vorbetrachtungen abgeschlossen wurden, können wir uns in diesem Kapitel dem zentralen Thema dieser Arbeit widmen: Der Berechnung des s, t -Zusammenhangspolynoms eines Graphen G mit Weglängenbeschränkung mittels Verwendung einer Baumzerlegung.

In Kapitel 2 wurde eine Methode eingeführt, um das s, t -Zusammenhangspolynom eines Graphen G mittels einer Baumzerlegung zu berechnen. Da hierbei die Komplexität für eine feste Weite w der Zerlegung polynomial in der Anzahl der Graphenknoten ist, soll die Weglängenbeschränkung D in die Methode integriert werden, um nach Möglichkeit die Güte der Komplexität beizubehalten.

Dazu werden im ersten Abschnitt Reduktionen des Graphen vorgestellt. Danach folgt eine erste Idee für ein Berechnungsverfahren unter Nutzung der in Kapitel 2 verwendeten Methode. Auf Grundlage der daraus entstehenden Probleme werden zusätzliche Strukturen in die Idee eingefügt. Zuletzt wird die Komplexität dieser Lösung bestimmt.

3.1 Reduktionen

Für die Anwendung der Baumzerlegung wurde unter anderem vorausgesetzt, dass der untersuchte Graph G schlicht sein soll. Durch den Einsatz von Reduktionen können auch Graphen zugelassen werden, die Schlingen und parallele Kanten besitzen.

Die Reduktion eines Graphen $G = (V, E, Pr(E))$ unter Weglängenbeschränkung D , also das Entfernen von speziellen Kanten und/oder Knoten im Graphen, ist ein wirksames Werkzeug, denn ein vereinfachter Graph vermindert den Berechnungsaufwand des DC2RP. Deswegen werden sie meist zu Beginn der Zuverlässigkeitsberechnung verwendet. Auch während der Berechnung ist ihre Anwendung denkbar. Wie und in welchem Umfang sie eingesetzt werden, liegt im Ermessen des Anwenders. Es sei angemerkt, dass die Reduktionen in Eigenleistung erstellt wurden. So wurde die Inzidenz-Reduktion aus [Can10] entnommen.

Bemerkung 3.1.1

- Die Entfernung von Kanten zieht keinerlei Veränderungen der Baumzerlegung mit sich. Die zweite Bedingung der Baumzerlegung ist die einzige, die sich auf die Kanten bezieht. Sie wird durch das Entfernen einer Kante nicht verletzt, da alle anderen Kanten bei unveränderter Baumzerlegung die zweite Bedingung weiterhin erfüllen.
- Sollten bei der Entfernung der Kanten isolierte Knoten entstehen, so sind letztere für D -Wege nicht mehr interessant und können entfernt werden. Es sei denn,

durch die Isolation dieses Knotens ist der s, t -Zusammenhang nicht mehr gewährleistet, dann war die Kantenentfernung fehlerhaft. Die Entfernung eines einzelnen Knotens v aus einer Baumzerlegung erfolgt durch Elimination des Knotens aus jedem Baumknoten (siehe Abbildung 3.1).

Die erste Bedingung der Baumzerlegung bleibt damit erfüllt. Die zweite Bedingung ebenso, da auch alle zu v inzidenten Kanten gelöscht werden. Für die dritte Bedingung nutzt man wieder die Teilbauminterpretation. Der v enthaltende Teilbaum wird entfernt. Es ergeben sich keine Auswirkungen auf die restlichen Teilbäume, da diese aus anderen Knoten bestehen. Die dritte Bedingung ist also weiterhin erfüllt.

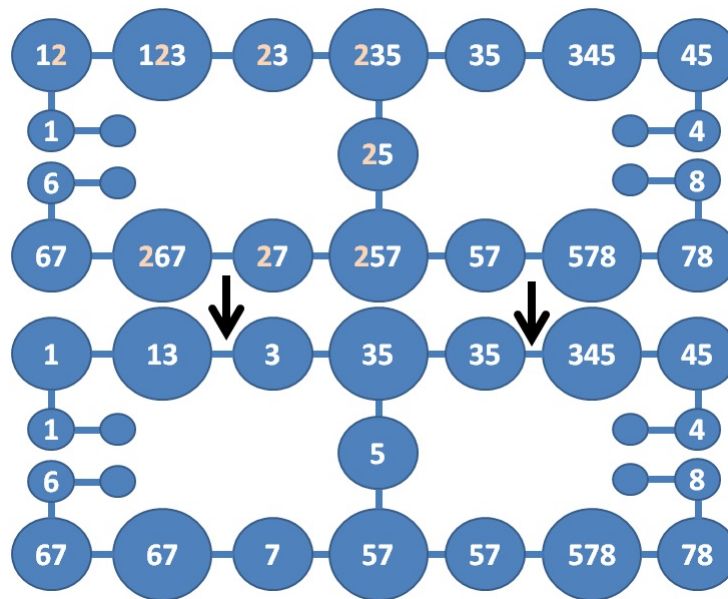


Abbildung 3.1: Entfernung des Knotens 2 aus der Baumzerlegung

Definition 3.1.1 (Kantenkontraktion)

Sei $e = \{u, v\}$ die Kante eines Graphen $G = (V, E)$, so bezeichnet G/e die **Kontraktion der Kante** e , also den durch Kontraktion von e aus G entstehenden Graphen. Dazu verschmelzen wir u und v zu einem neuen Knoten w , der zu allen Knoten benachbart ist, zu denen zuvor u oder v benachbart waren. u und v sollen keine parallelen Kanten zueinander besitzen.

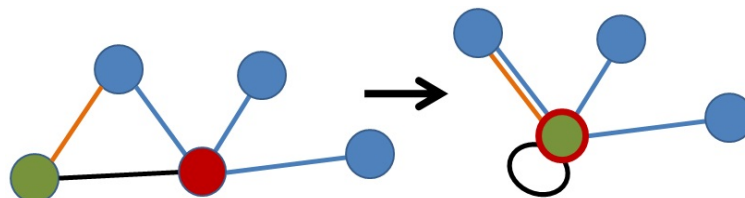


Abbildung 3.2: Umsetzung der Kantenkontraktion

- Die Kantenkontraktion wird in der Baumzerlegung realisiert, indem wir u und v durch w ersetzen (siehe Abbildung 3.3).

Wie man leicht sieht, ist damit die erste Bedingung der Baumzerlegung weiterhin erfüllt. Ebenso die zweite Bedingung: alle Kanten mit Endknoten u oder v haben nun Endknoten w . Da vorher die zweite Bedingung erfüllt war, ist sie nun durch Umbenennung von u, v zu w weiterhin erfüllt. Die dritte Bedingung lässt sich an Hand ihrer Interpretation nachweisen: Der Teilbaum aller u enthaltenden Baumknoten wird durch die Umbenennung zum Teilbaum der w enthaltenden Baumknoten. Das gilt analog für v . Da die Kante $e = \{u, v\}$ kontrahiert wurde, existiert nach der zweiten Bedingung ein Baumknoten, der u und v enthält. Damit überlagern sich beide Teilbäume. Folglich entstehen bei der Umbenennung keine zwei getrennten Teilbäume, sondern ein größerer, zusammenhängender. Damit ist die dritte Bedingung erfüllt.

Durch die Kontraktion können identische Baumknoten entstehen. Sind zwei identische Baumknoten benachbart und wenigstens einer ist kein Join oder Blatt, so können wir diese Baumknoten kontrahieren.

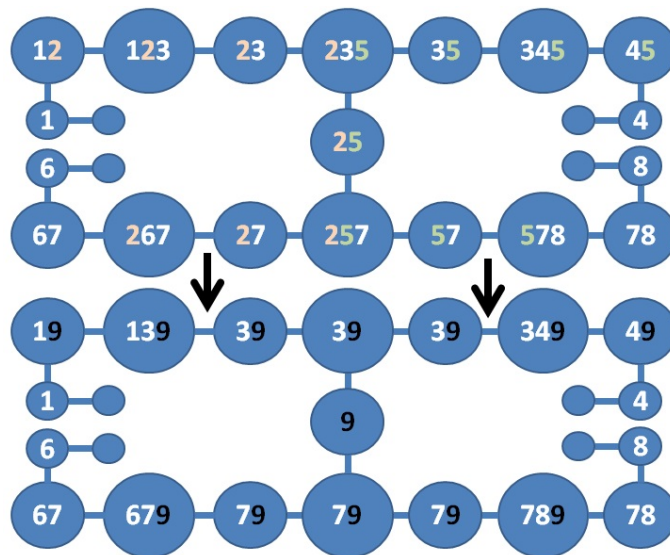


Abbildung 3.3: Umsetzung der Kontraktion in der Baumzerlegung

3.1.1 Schlingen-Reduktion

Der Untergraph G' geht aus G durch Entfernung aller Schlingen hervor. Schlingen können ihrer Definition wegen kein Teil eines D -Weges sein. Darum wirkt sich ihre (Nicht-) Existenz nicht auf den Zusammenhang aus.

Für Schlingen ist die zweite Bedingung der Baumzerlegung auf Grund ihrer Definition bereits mit der ersten Bedingung erfüllt. Damit sind keine Änderungen an der Baumzerlegung vorzunehmen.

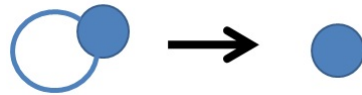


Abbildung 3.4: Schlingen-Reduktion

3.1.2 Brücken-Reduktion

Beinhaltet der Graph eine Brücke e (oder mehrere), die Element aller D -Wege ist, so muss sie intakt sein, um den s, t -Zusammenhang zu gewährleisten. Damit kann die Brücke kontrahiert und die Weglängenbeschränkung um eins vermindert werden:

$$R_{s,t}(G, D) = p_e R_{s,t}(G/e, D - 1)$$

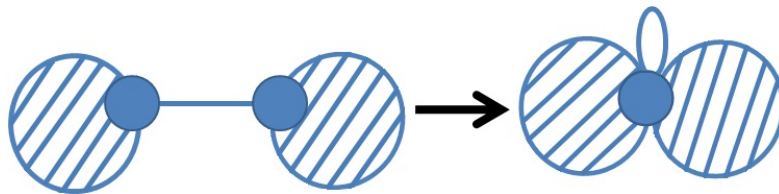


Abbildung 3.5: Brücken-Reduktion

3.1.3 Parallel-Reduktion

Angenommen es existieren zwischen zwei Knoten u, v mehrere parallele Kanten. Dann ist die Wahrscheinlichkeit, dass die Knoten verbunden sind, gleich der Wahrscheinlichkeit, dass mindestens eine Kante intakt ist. Mit der Unabhängigkeit der Kantenausfälle folgt, dass wir alle Kanten löschen und eine neue Kante e mit der Wahrscheinlichkeit

$$p_e = 1 - \prod_{f \in P_{u,v}(1)} q_f \quad (3.1)$$

einfügen können. Eine Änderung der Baumzerlegung ergibt sich nicht. Ist die zweite Bedingung für eine der parallelen Kanten von u, v erfüllt, so auch für alle anderen.

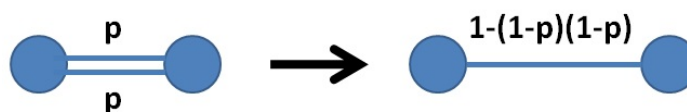


Abbildung 3.6: Parallel-Reduktion

3.1.4 Weg-Reduktion

Gesetzt den Fall, dass s, t nur über zueinander disjunkte D -Wege P_1, \dots, P_k verbunden sind, kann eine allgemeinere Form der Parallel-Reduktion verwendet werden. In (3.1) wird die Ausfallwahrscheinlichkeit einer Einzelkante durch die Wahrscheinlichkeit des Gegenereignisses ersetzt, dass alle Kanten eines Weges intakt sind (fielen eine der Kanten aus, wäre der Weg nicht intakt).

$$R_{s,t}(G, D) = 1 - \prod_{i=1}^k \left(1 - \prod_{e \in P_i} p_e \right)$$

Angenommen, alle Wege besitzen die gleiche Länge h . Anfangs- und Endknoten seien aber zwei Artikulationen des Graphen. Und zwar derart, dass das Ersetzen der Wege durch eine Einzelkante zu einer Brücke führt, die Teil eines D -Weges ist. Dann kann die Weg-Reduktion mit einer Brücken-Reduktion kombiniert werden:

$$R_{s,t}(G, D) = \left(1 - \prod_{i=1}^k \left(1 - \prod_{e \in P_i} p_e \right) \right) R_{s,t}(G - \{e \in P_i, i \in \{1, \dots, k\}\}, D - h)$$

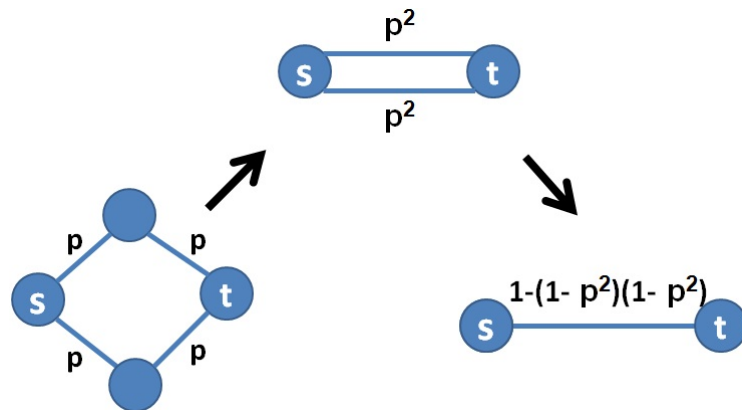


Abbildung 3.7: Weg-Reduktion

Dieser Vorgang lässt sich durch eine Mehrfachausführung von Kantenkontraktion und Parallel-Reduktion realisieren. Entsprechend wird die Weg-Reduktion in der Baumzerlegung umgesetzt.

3.1.5 Artikulations-Reduktion

Gibt es Artikulationen im Graphen, so lassen sich diese in zwei Arten kategorisieren:

1. Sie unterteilen den Graphen in zwei Untergraphen, von denen der eine s und der andere t enthält. Jeder s, t -Weg muss dementsprechend durch diesen Knoten laufen und jeweils eine zu ihm inzidente Kante im einen und im anderen Untergraphen besitzen.
2. Sie unterteilen den Graphen in zwei Untergraphen, von denen einer s und t enthält und der andere nicht. Ein s, t -Weg liegt somit nur in einem der beiden Untergraphen.

Der Verlauf eines kürzesten s, t -Weges lässt sich damit als Indikator verwenden.

Untergraphen an Artikulationen zweiter Art die s und t nicht enthalten, können auf Grund der Eigenschaft eines D -Weges (Vermeidung mehrfachen Durchlaufens von Knoten) komplett entfernt werden.

Oben benannte zu entfernende Untergraphen können also knotenweise aus der Baumzerlegung entfernt werden. Identische adjazente Baumknoten kann man wieder kontrahieren. Leere Baumknoten, die keine Blätter sind, können dabei nicht zurückbleiben. Ansonsten müsste dies als Zerlegung des Graphen in Komponenten interpretiert werden und das geschieht bei dieser Reduktion nicht.

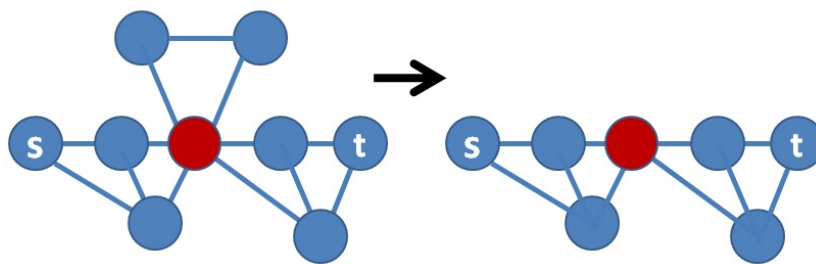


Abbildung 3.8: Artikulations-Reduktion

3.1.6 Inzidenz-Reduktion

Diese Reduktion setzt voraus, dass alle zum Knoten t inzidenten Kanten intakt sind (Ausfallwahrscheinlichkeit 0). Erreicht ein D -Weg einen Nachbarknoten von t , so erreicht dieser Weg mit Wahrscheinlichkeit 1 den Knoten t . Alle Wege der Länge kleiner gleich $D - 1$, die diese Nachbarknoten nicht erreichen, würden auch als Wege mit Länge kleiner gleich D den Knoten t nicht erreichen. Es brauchen somit nur die Wege der

Länge kleiner gleich $D - 1$ untersucht zu werden. Um alle Algorithmen weiterhin ordnungsgemäß anwenden zu können, kann t und seine Nachbarschaft zu einem Knoten kontrahiert und alle Schleifen gelöscht werden; der entstandene Graph sei G' .

$$R_{s,t}(G, D) = R_{s,t}(G', D - 1)$$

Die Inzidenz-Reduktion lässt sich auch für s und Artikulationen der ersten Art anwenden. Die Umsetzung in der Baumzerlegung erfolgt durch mehrfache Kantenkontraktion.

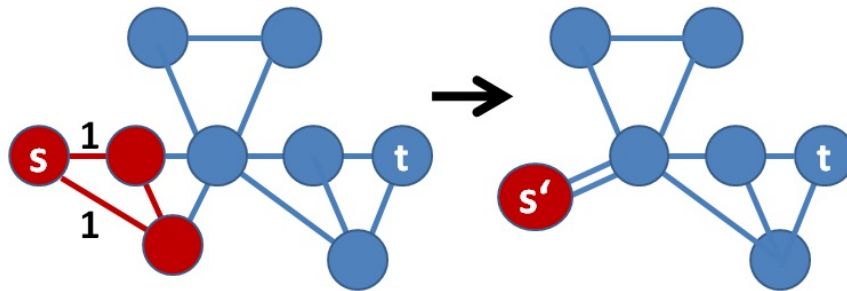


Abbildung 3.9: Inzidenz-Reduktion

3.1.7 Reduktion irrelevanter Kanten

Breitensuche

Alle Kanten, die keinem D -Weg angehören, sind irrelevant, da ihre Intaktheit nicht von Belang ist. Sie können aus dem Graphen entfernt werden. Zu diesem Zweck wird standardmäßig eine **Breitensuche** (kurz: BFS, engl.: *breadth-first-search*) durchgeführt, um die Längen kürzester Wege in einem zusammenhängenden, ungerichteten, ungewichteten Graphen $G = (V, E)$ zu berechnen. Dies geschieht folgendermaßen:

1. Wähle einen Startknoten s , dieser erhält die Kennung 0.
2. Bilde die Knotenmenge $U = N(v)$. Alle darin enthaltenen Knoten erhalten die Kennung 1. Bilde K als Menge aller Knoten mit Kennung.
3. Bilde die Knotenmenge $U = N(U) \setminus K$. Alle Knoten ohne Kennung erhalten die Kennung 2. Aktualisiere K .
4. Wiederhole Schritt 3, bis alle Knoten eine Kennung erhalten haben. Für die n -te Wiederholung von Schritt 3 wird die Kennung $n + 2$ verwendet.

Die Kennung eines Knotens v entspricht $d(s, v)$ (vergleiche dazu [Internet03]). Die Laufzeit liegt in $O(|V| + |E|)$.

Methode

Eine selbstentworfenene Methode zur Reduktion irrelevanter Kanten arbeitet in drei Stufen:

1. Ermittle über eine Breitensuche den minimalen Abstand jedes Knotens von s und schreibe diesen in die Knoteninformation d_s .
2. Ermittle über eine Breitensuche den minimalen Abstand jedes Knotens von t und schreibe diesen in die Knoteninformation d_t .
3. Entferne jede Kante $\{u, v\}$, $u, v \in V$ mit $d_s(u) + d_t(v) + 1 > D$ (der Weg von s zum Knoten u über die Kante und vom Knoten v nach t ist zu lang, kann also nicht zu einem D -Weg gehören).

Bemerkung 3.1.2

Der Schwerpunkt dieses Algorithmus liegt nicht auf den Knoten. Beispielsweise wird beim Diamantgraph und Beschränkung $D = 2$ kein Knoten entfernt. Die Kante der adjazenten Knoten u und v mit $\deg(u) = \deg(v) = 3$ gehört zu keinem D -Weg und ist deshalb reduzierbar (siehe Abbildung 3.10).

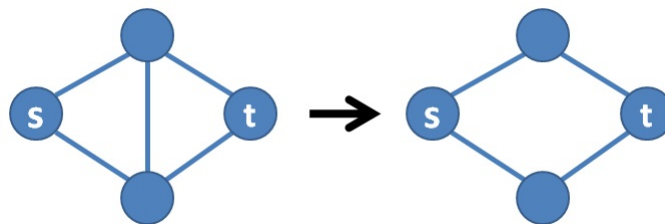


Abbildung 3.10: Reduktion der irrelevanten Kante im Diamantgraph

Algorithmus zur Reduktion irrelevanter Kanten

Parameter:

- G Graph $G = (V, E)$
 s, t Knoten deren Zusammenhang ermittelt werden soll
 N Knotenmenge
 $U(N)$ Menge der unmarkierten Nachbarknoten der Knotenmenge N
 (mit Ausnahme t)
 $d_s(k)$ Abstand des Knotens k von s
 $d_t(k)$ Abstand des Knotens k von t

Algorithm 2 Algorithmus zur Reduktion irrelevanter Kanten

reduktion(G, s, t):**Eingabe:** zu reduzierender Graph G **Ausgabe:** reduzierter Graph G' $d_s(V) = \infty$ {initialisiere alle Knoten} $d_t(V) = \infty$ $N := s$ {initialisiere Knotenmenge} $d_s(s) = 0$ **for** $i = 1$ **to** $i < D + 1$ **do** $d_s(U(N)) := i$ {weise jedem unmarkierten Nachbarknoten der markierten Knotenmenge den Wert (und damit Abstand von s) i zu} $N := U(N) + N$ {die jetzt neu markierten Knoten werden in Menge der markierten Knoten aufgenommen}**end for**{nachfolgend gibt es keinen D -Weg}**if** $d_s(\text{aller Nachbarknoten von } t) == \infty$ **then**Lösche V, E **return** G **end if**entferne alle Knoten mit Markierung ∞ und die zu ihnen inzidenten Kanten $N := t$ {initialisiere Knotenmenge} $d_t(t) = 0$ **for** $i = 1$ **to** $i < D + 1$ **do** $d_t(U(N)) := i$ {weise jedem unmarkierten Nachbarknoten der markierten Knotenmenge den Wert (und damit Abstand von t) i zu} $N := U(N) + N$ {die jetzt neu markierten Knoten werden in Menge der markierten Knoten aufgenommen}**end for**{neue Kantenmenge E' }**for all** $\{u, v\}$ mit $u, v \in V$ **do****if** $\min(d_s(u) + d_t(v) + 1, d_s(v) + d_t(u) + 1) > D$ **then**Lösche $\{u, v\}$ **end if****end for**{neue Knotenmenge V' }**for all** isolierte Knoten v in V **do**Lösche v **end for****return** $G'(V', E')$

Komplexität des Algorithmus

1. Schritt: Breitensuche von t aus: $O(m)$
 2. Schritt: Wiederholung des ersten Schrittes: $O(m)$
 3. Schritt: Betrachtung jeder Kante: $O(m)$
- zum 3. Schritt: Überprüfung jedes Knotens: $O(n)$
- Gesamt:** $O(m + n)$

Mit diesem Algorithmus kann nicht ausgeschlossen werden, dass ein kürzester Weg von s nach u und von v nach t Knoten doppelt benutzt ($u, v, s, t \in V$, $(u, v) \in E$). Schritt 3 schließt durch $\min(d_s(u) + d_t(v) + 1, d_s(v) + d_t(u) + 1) > D$ zumindest aus, dass bei solchen kürzesten Wegen die Kante $\{u, v\}$ doppelt abgelaufen wird.

Wenn $d_s(u) = d_s(v)$ und/oder $d_t(u) = d_t(v)$, folgt, dass sich u und v von s und/oder t auf zwei verschiedenen kürzesten Wegen erreichen lassen. Damit wird bei Schritt 3 ein Weg ausgeschlossen, der $\{u, v\}$ doppelt verwendet.

Wenn $d_s(u) = d_s(v) + 1$ und $d_t(v) = d_t(u) + 1$ (oder umgekehrt), dann wird durch Schritt 3 die Variante gewählt, die $\{u, v\}$ nicht benutzt.

Wenn $d_s(v) = d_s(u) + 1$ und $d_t(v) = d_t(u) + 1$, dann ergeben sich folgende zwei Fälle:

1. Es gibt D -Wege, die die Knoten s, u, v, t in dieser Reihenfolge ablaufen, ohne u oder v wiederholt zu benutzen. Diese Wege müssen nicht notwendigerweise kürzeste Wege sein. In diesem Fall gibt es keine Probleme.
2. Es existiert kein D -Weg, der die Kante $\{u, v\}$ benutzt, ohne sie zweimal verwenden zu müssen (siehe Abbildung 3.11). Knoten u muss also vor v benutzt werden und von v aus existiert kein Weg nach t , ohne $\{u, v\}$ abzulaufen oder die Weglängenbeschränkung D einzuhalten. Damit ist u entweder eine Artikulation zweiter Art und $\{u, v\}$ wird durch die Artikulations-Reduktion entfernt oder (und damit wird auf den zweiten Teil des vorherigen Satzes Bezug genommen) es gibt eindeutige irrelevante Kanten, die eine Charakterisierung von u als Artikulation verhindern. Diese sind nach Abschluss des Algorithmus zu entfernen, u lässt sich als Artikulation identifizieren und $\{u, v\}$ wird ebenfalls entfernt.

Diese Überlegungen gelten auch speziell für kürzeste Wege. Eine einmalige Verwendung dieser Reduktion entfernt nicht alle irrelevanten Kanten, aber bei mehrfacher Ausführung und in Kombination mit anderen Reduktionen führt sie zum gewünschten Ergebnis.

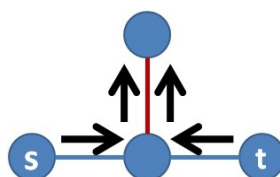


Abbildung 3.11: Beispiel für eine doppelt abgelaufene Kante (rot)

3.1.8 Serien-Reduktion

Angenommen, im Graphen existieren nach obigen Reduktionen drei Knoten $u, v, w \in V$ mit $N(v) = \{u, w\}$. So existiert ein Weg von u nach w über v nur dann, wenn die Kanten $\{u, v\}$ und $\{v, w\}$ intakt sind. Durch Kontraktion lässt sich dann eine Kante $\{u, w\}$ mit folgender Wahrscheinlichkeit erzeugen:

$$P_{u,w} = P_{u,v}P_{v,w}$$

Diese Kante steht hier für einen Weg der Länge 2. Im Gegensatz zu obigen Reduktionen wie der Brücken- oder der Weg-Reduktion, lässt sich diese topologische Verkürzung nicht auf die Weglängenbeschränkung D verlagern, da die dafür notwendigen Voraussetzungen nicht erfüllt sind.

Lösung: Der Graph G wird gewichtet, indem die Kante $\{u, w\}$ folgendes Kantengewicht erhält:

$$\text{weight}(\{u, w\}) = \text{weight}(\{u, v\}) + \text{weight}(\{v, w\})$$

Das Kantengewicht entspricht der Anzahl an Kanten, die durch $\{u, w\}$ dargestellt werden. Damit ist eine Reduktion von nicht verzweigenden Wegen möglich, die nicht den Bedingungen der Weg-Reduktionen entsprechen.

Durch den Einsatz von Reduktionen entstehen gewichtete Kanten. Der Umgang mit ihnen wird in Kapitel 3.22 beschrieben.



Abbildung 3.12: Serien-Reduktion

Probleme der Serien-Reduktion:

Sind, ähnlich wie bei der Weg-Reduktion, zwei Knoten u, v durch mehrere Wege verbunden, so werden die Wege zu parallelen Kanten reduziert. Diese könnten mittels Parallel-Reduktion zu einer einzigen zusammengefasst werden. Die Kantengewichte stellen dabei ein Problem dar. Besitzen alle reduzierten Kanten dasselbe Gewicht, so ist die Weglänge bei Wahl einer der Kanten in jedem Fall dieselbe. Sie können also mittels einer Parallel-Reduktion zusammengefasst werden. Die neue Kante erhält das gemeinsame Gewicht der reduzierten Kanten.

Unterscheiden sich die Kantengewichte aber, so lässt sich nach einer Parallel-Reduktion keine eindeutige Weglänge und damit kein eindeutiges Gewicht festlegen. Man darf die Kanten dann nicht reduzieren. Als Folge dessen würden parallele Kanten existieren. In der DC2RP-Berechnung könnte dies nicht erfasst werden.

Um derartige Fälle zu vermeiden, muss eine weitere Bedingung an die Serien-Reduktion gestellt werden: Wenn durch eine Serien-Reduktion zwei parallele Kanten mit unterschiedlichen Gewichten existieren, so ist die Reduktion nicht durchzuführen.

3.1.9 Hinweis zur Anwendung

In Kapitel 2 besitzen alle Kanten des bearbeiteten Graphen G die Wahrscheinlichkeit p . Durch Reduktionen werden Kanten e mit Wahrscheinlichkeit $p_e \neq p$ erzeugt. Demzufolge ist während der Zuverlässigkeitsberechnung bei besagten Kanten p_e und $1 - p_e$ zu verwenden.

Durch einige Reduktionen entstehen gewichtete Kanten. Wir untersuchen somit nicht nur ungewichtete, sondern auch mit gewichtete Graphen. Wie später zu sehen sein wird, arbeiten wir ohnehin mit Weglängen und nichts anderem entsprechen die Gewichte. Damit ergeben sich keine Unterschiede zum ungewichteten Fall, bis auf die Kantenuntersuchung.

Weitere Reduktionen sind denkbar, auch wenn nicht alle von ihnen auf Grund der nötigen Identifizierbarkeit von Weglängen anwendbar sind.

3.2 Einführende Idee

Wir werden nun versuchen, das in Kapitel 2 vorgestellte und in Kapitel 2.4 modifizierte Verfahren zur Berechnung des s, t -Zusammenhangspolynoms um die Weglängenbeschränkung zu erweitern. Daraus ergeben sich zusätzlich benötigte und zu speichernde Informationen. Notwendigerweise sind dadurch die Zustände zu ändern. Ihre Korrektheit wird anhand der Funktionalität der vier Grundschritte überprüft und die Zustände bei auftretenden Problemen entsprechend angepasst.

3.2.1 Ausgangsüberlegungen

Der Ausgangspunkt der Überlegungen ist die Definition des DC2RP, welches eine Summe von Polynomen ist. Jedes Polynom für sich gibt die Eintrittswahrscheinlichkeit eines zulässigen Untergraphen von G an. Ein Untergraph von G ist genau dann zulässig, wenn er einen D -Weg besitzt.

Während der Berechnung werden Teilgraphen konstruiert. Es ist damit nicht möglich, während eines beliebigen Schrittes der Berechnung in polynomialer Zeit zu sagen, welche Untergraphen sich aus dem bisher untersuchten Graphen ergeben. Könnte man dies, so brauchte man lediglich alle Untergraphen zu konstruieren und die unzulässigen zu entfernen. Dieses Vorgehen wächst jedoch exponentiell in der Kantenzahl von G .

Allerdings ist es möglich, eine andere Aussage über einen Teilgraphen zu treffen: Entwickeln sich aus dem Teilgraphen keine zulässigen Untergraphen, so ist der Teilgraph in der weiteren Berechnung nicht mehr zu beachten.

Dies ist genau dann der Fall, wenn es keinen D -Weg gibt. Dazu müsste $d(s, t)$ über

eine Breitensuche bestimmt werden. Wir können nicht alle Kanten für jeden Teilgraphen speichern. Dies würde zur Konstruktion aller Untergraphen führen. Der Speicheraufwand ist also zu reduzieren. Bisher wurde dazu der Zusammenhang der Frontknoten als Partitionen gespeichert. Wir konnten damit mehrere Teilgraphen zusammenfassen.

3.2.2 Die Idee

Man betrachte einen Schritt i der Berechnung. O.B.d.A. gehen wir davon aus, dass s bereits aktiviert wurde und t noch nicht. Allen Teilgraphen ist derselbe nicht untersuchte Teil des Graphen G gemein. Sie unterscheiden sich erst ab den Frontknoten. Um den Speicheraufwand zu reduzieren, müssen die Informationen der Teilgraphen dahingehend eingeschränkt werden, dass man Knoten und Kanten der Teilgraphen entfernt. Zugleich muss die Ermittlung von $d(s, t)$ weiterhin gewährleistet sein.

Dazu kann man die gewohnte Einteilung der Teilgraphen in Zustände (π, P_π) von Partition und Eintrittswahrscheinlichkeit verwenden. Die Zustände sind zu erweitern. Da zusätzlich der Abstand der Frontknoten zu s benötigt wird, muss man diesen für jeden Frontknoten mitspeichern; alle Abstände der Frontknoten eines Teilgraphen werden in noch unbestimmten Strukturen A gespeichert. In einer Partition sind allerdings mehrere Teilgraphen zusammengefasst. Damit sind in jedem Knoten einer Partition so viele Strukturen A gespeichert, wie es Teilgraphen mit unterschiedlichen Frontknotenabständen gibt. Das Polynom eines Zustands wird auf diese gespeicherten Strukturen A verteilt. Die Partitionen dienen lediglich der Zusammenfassung aller Strukturen A , deren Frontknoten denselben Zusammenhang besitzen.

Ist dies realisiert, so kann jederzeit eine Breitensuche von t aus gestartet werden. Dieser bestimmt den Abstand der Frontknoten von t . Da alle Zustände dieselben Frontknoten und denselben nicht untersuchten Graphenteil von G besitzen, ist dieser Vorgang nur einmal durchzuführen. Danach werden die Zustände bzw. die gespeicherten Abstände der Partitionen auf $d(s, t) \leq D$ hin untersucht. Dabei wird in jedem A nach einem Knoten v gesucht, bei dem $d(s, v) + d(v, t) \leq D$ gilt. Da der erste Abstand ein gespeicherter Wert und der zweite einmalig linear in der Anzahl der nicht untersuchten Knoten ist, lässt sich eine solche Überprüfung schnell realisieren. Alle Strukturen A , die keinen geeigneten Knoten besitzen, können mitsamt ihres Polynoms entfernt werden.

3.2.3 Vorteile

Die Idee der Abstandsspeicherung lässt sich scheinbar gut in den vier Grundschritten integrieren. Die Abstände der Strukturen A sind dabei wie Partitionen zu behandeln:

Aktivierung:

Ein neuer Knoten wird in allen Partitionen (und ihren Strukturen A) als Einzelblock mit Abstand unendlich aufgenommen.

Kantenuntersuchung:

Bei Einfügen einer neuen Kante werden die Abstände der beiden verbundenen Knoten verglichen. Der größere Abstand wird durch den des kleineren ersetzt und eins addiert. So erhalten auch Knoten mit Abstand unendlich einen endlichen Wert. Zur Verbindung zweier Blöcke siehe Nachteile (*Blöcke mit Einträgen unendlich*).

Deaktivierung:

Die Deaktivierung erfolgt wie üblich. Jedoch werden die Strukturen A einer Partition zusammengefasst, wenn ihre Abstände identisch sind.

Fusion:

Dieser Schritt erfolgt wie sonst auch durch Kombination aller Strukturen A beider Teilbäume. Besitzt ein Knoten nach Supremumbildung unterschiedliche Abstände, so wird der kleinere übernommen.

3.2.4 Nachteile

Leider ist die in den Vorteilen beschriebene Einfügung der Abstandsspeicherung mangelhaft. Zudem wirft die Abstandsberechnung selbst einige Fehler auf:

- *Blöcke mit eingetragenen Werten unendlich:*

Die einzigen Knoten, die einen endlichen Abstand zugewiesen bekommen, sind die Knoten des s -Blocks. Wird ein Block mit dem s -Block, z.B.: durch Kantenuntersuchung, vereinigt, so erhält nur der Endknoten der Kante einen endlichen Wert zugewiesen. Die restlichen Knoten behalten ihre unendlichen Werte. Analog bei der Fusion. Es fehlen also Informationen über den Zusammenhang bzw. den Abstand dieser restlichen Knoten zu s .

Eine Alternative dazu wäre die Speicherung der kompletten Teilgraphen, die durch einen unmarkierten Block dargestellt werden. Beim Verbinden mit dem markierten Block können damit die Abstandsinformationen gewonnen und die Nicht - Frontknoten gelöscht werden. Wie man sich leicht überlegen kann, ist diese Variante speicherintensiv und damit unpraktikabel.

- *Ungenügende Breitensuche:*

Angenommen, es läge der Teilgraph der Abbildung 3.13 mit einem kürzesten s, t -Weg (schwarz) vor:

Dabei bezeichnet der blaue Anteil den nicht untersuchten Teil des Graphen und der rote/orange einen Teilgraphen. Betrüge die Weglängenbeschränkung $D = 8$, so würde die Abstandsberechnung per Breitensuche über nicht untersuchte Kanten zu einem Ausschluss dieses Teilgraphen (bzw. seiner Struktur A) führen. Dabei existiert ein zulässiger s, t -Weg, der nicht erfasst wird, weil er bereits untersuchte Kanten und deaktivierte Knoten nutzt und durch zwei Blöcke verläuft. Eine geeignete Abstandsberechnung wird benötigt.

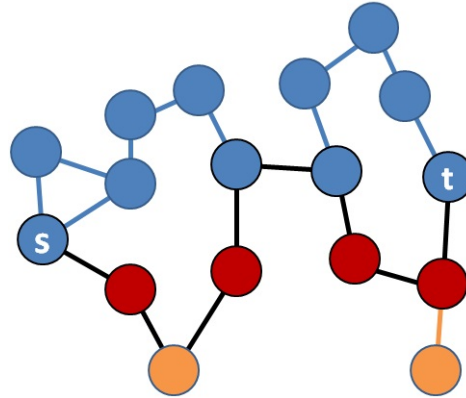


Abbildung 3.13: Weg (schwarz) über zwei Blöcke (rot, orange für deaktivierte Knoten)

- *Neue Abstände durch kürzeste Wege:*
Angenommen, es liegt der Teilgraph der Abbildung 3.14 vor:

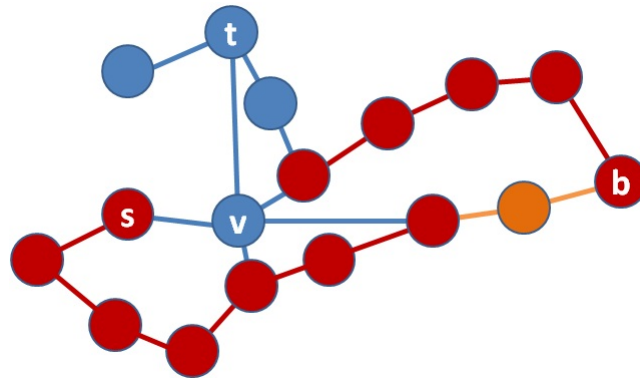


Abbildung 3.14: Problem fehlerhafter kürzester Wege

Durch die Aktivierung des Knotens v und Einfügen der Kanten zu den restlichen Frontknoten entsteht ein neuer kürzester Weg von s nach b . Da der orange Knoten schon deaktiviert wurde, werden er bei der Abstandsberechnung nicht mit einbezogen. Damit kann nur die Viererkette zwischen v und b für die kürzeste-Weg-Berechnung herangezogen werden. Dadurch wird b ein falscher Wert $d(s, b)$ zugewiesen. Es sind also weitere Informationen über den Zusammenhang zwischen den Frontknoten nötig.

Neben den vorgestellten Nachteilen sind sicherlich noch weitere zu finden. Die Obigen dienen bei der Idee zur Charakterisierung der zu realisierenden Schwerpunkte in der Abstandsspeicherung:

- *Feststellung der Entfernung in den unmarkierten Blöcken*
- *Erstellung einer schnellen Abstandsberechnung, welche auch Wege durch mehrere Blöcke zulässt*
- *Realisierung neuer Abstände auf Grund neuer kürzester Wege durch zusätzliche Informationen über den Zusammenhang*

3.3 Einführung der Abstandsmatrizen

Betrachtet man die Nachteile der ersten Idee einmal genauer, so ist festzustellen, dass ihr gemeinsamer Fehler in den mangelnden Informationen über den Zusammenhang der Frontknoten besteht. Wüssten wir, wie groß die Abstände zwischen den Knoten der unmarkierten Blöcke sind, so wäre es ein Leichtes, ihren Abstand zu s bei Verbund mit einem markierten Block zu ermitteln. Auch ein Weg über mehrere Blöcke wäre damit möglich. Ebenso besäßen wir mehr verwertbare Informationen bei Entstehung neuer kürzester Wege.

Dem Vorschlag des Zweitprüfers und Betreuers M. Dod folgend, erweitern wir die oben eingeführte Idee: Neben den Abständen zu s werden zusätzlich die Abstände zwischen den Frontknoten gespeichert.

Wie man sich leicht überlegen kann, stellen Matrizen dabei ein günstiges Speichermedium dar:

Definition 3.3.1 (Abstandsmatrix)

Gegeben sei ein Graph $G = (V, E)$. Sei $X \subseteq V$, mit $|X| = x$. Dann ist die **Abstandsmatrix** A der Knotenmenge X folgendermaßen definiert:

$$A = (a_{i,j})_{x,x} \text{ mit } a_{i,j} = d(i, j), \forall i, j \in X$$

mit $d(i, j) = \infty$ falls i, j in verschiedenen Komponenten liegen.

Man betrachte sich einen Schritt i der Zuverlässigkeitsberechnung:

Dann entspricht X der Menge der Frontknoten X_i inklusive s' und/oder t' (insofern s und /oder t schon deaktiviert worden sind). Der Graph G der Definition stellt dabei einen Teilgraphen G^i der DC2RP-Berechnung dar, für den die Abstände der Frontknoten in die Matrix A eingetragen werden (siehe Abbildung 3.15).

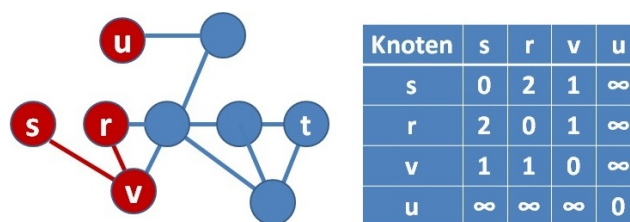


Abbildung 3.15: Beispiel einer Abstandsmatrix; G besteht aus den roten Knoten

Folgerung 3.1

Es soll ausgenutzt werden, dass Einträge $a_{i,j}$ genau dann einen festen Wert besitzen, wenn die Knoten i und j in derselben Komponente liegen. Dadurch ist es stets möglich, A in eine Blockdiagonalmatrix umzuformen. Man schreibe die Spaltenindizes dazu derart um, dass sich zwischen je zwei Knoten derselben Komponente kein Knoten einer anderen Komponente befindet. Die Reihenfolge der Zeilenindizes soll gleich der Reihenfolge der Spaltenindizes sein (siehe Abbildung 3.15).

Folgerung 3.2

A liege als Blockdiagonalmatrix vor. Alle Knoten eines Blockes aus A sind zusammenhängend in G^i . Vergleicht man diese Blöcke mit den Blöcken der Partition π von G^i , so ist auf Grund des Zusammenhangs der Blockknoten leicht einzusehen, dass A in π überführt werden kann (siehe Abbildung 3.16).

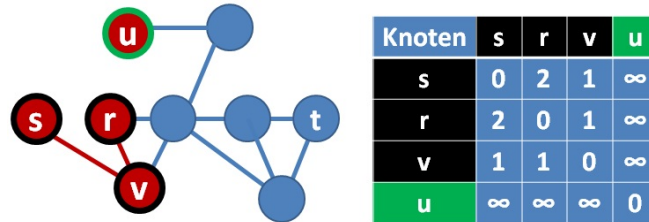


Abbildung 3.16: Schwarz: Komponente/Block 1; Grün: Komponente/Block 2; Partition: $suv|r$

Bemerkung 3.3.1

Mit den zusätzlichen Informationen über die Abstände der Frontknoten kann man an Stelle der Partition eines Teilgraphen G^i seine Abstandsmatrix A verwenden.

Folgerung 3.3

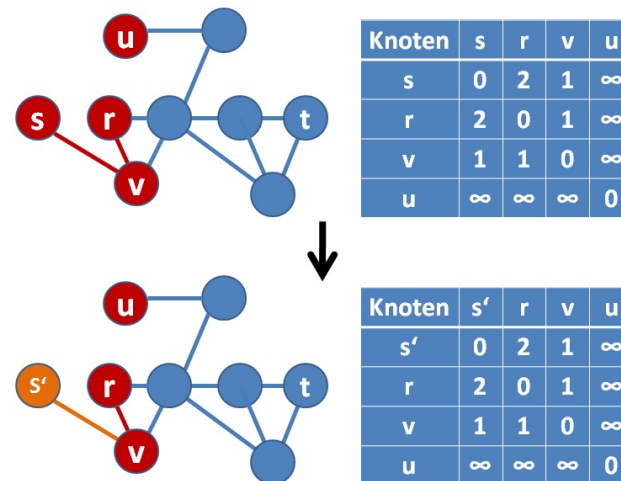
Partitionen fassen in Zuständen mehrere Teilgraphen zusammen, deren Frontknoten denselben Zusammenhang besitzen. Auf Grund der vorherigen Folgerung können also anstatt Partitionen Abstandsmatrizen in den Zuständen verwendet werden. Allerdings mit folgender Einschränkung:

Zwei Teilgraphen können nur dann in einer Abstandsmatrix zusammengefasst werden, wenn ihre Frontknoten **denselben Zusammenhang** besitzen und ihre **Abstände übereinstimmen**.

Die zusätzlich zu speichernden Informationen können bei einer Partition zu mehreren Abstandsmatrizen führen und erhöhen dadurch die Anzahl an Zuständen. Um den Bezug zu Kapitel 2 zu erhalten und eine bessere Vorstellung zu ermöglichen, sollen in einem Zustand sowohl Partition als auch Abstandsmatrix geführt werden.

Folgerung 3.4

Die Berechnung der weglängenbeschränkten s, t -Zusammenhangswahrscheinlichkeit benötigt die Markierung von s und/oder t enthaltenden Blöcken. Eine solche wurde bereits in Kapitel 2.4 eingeführt. Erschien sie zu jenem Zeitpunkt noch als aufwendiger als die bloße Markierung durch Unterstreichung, so erweist sie sich nun als günstig. Durch die Existenz von s und/oder t in einem Block wird dieser nicht nur markiert, sondern zugleich die Abstände der Blockknoten zu s und/oder t in A für weitere Berechnungen gespeichert und aktualisiert, auch wenn s und t deaktiviert wurden (siehe Abbildung 3.17).

Abbildung 3.17: Deaktivierung des Knotens s **Folgerung 3.5**

Die Zustände als Informationsspeicher bleiben erhalten, ebenso wie das Umfeld in dem sie definiert sind. Lediglich die Abstandsmatrizen sind darin zu integrieren:

Für eine Teilmenge $X \subseteq V$ sei $\mathbb{P}(X)$ die **Menge aller Abstandsmatrizen von X** . Sei X_i die aktive Menge der Front in Schritt i inklusive s' und/oder t' (insofern s und/oder t deaktiviert wurden). Jeder neue Schritt in der Baumzerlegung erzeugt eine Menge von Zuständen. In diesem Kontext ist ein **Zustand** ein Tripel (π_A, A, P_A) , wobei $A \in \mathbb{P}(X_i)$, π_A zu A gehörige Partition und $P_A = P(G^i, A)$. Hierbei beschreibt G^i den bis zum Schritt i erzeugten Teilgraphen. Sei Z_i die **Menge aller Zustände in Schritt i** . $(\{\emptyset\}, \emptyset, 1)$ mit leerer Partition, leerer Matrix und Polynom 1 wird als Startzustand in allen Blättern (außer der Wurzel) initialisiert.

3.4 Anwendung der Abstandsmatrizen in der Zuverlässigkeitsberechnung

In diesem Abschnitt wollen wir die Abstandsmatrizen in die vier Grundschritte für den Schritt i integrieren. Das in Kapitel 2 benutzte Bottom-Up-Verfahren bleibt erhalten, ebenso wie die Bedingungen für die Verwendung jedes Grundschrittes und die in Kapitel 2.4 eingeführte Erweiterung für Markierungen.

3.4.1 Aktivierung eines Knotens v

Dieser Schritt ist der einfachste der vier Grundschritte. v wird wie gewohnt als Einzelblock in die Zustände aufgenommen. Für A heißt dies, dass eine neuen Spalte v und eine neuen Zeile v in der Abstandsmatrix $A_{x,x}$ angelegt werden, so dass der Eintrag

$a_{v,v} = 0$ und die restlichen Einträge unendlich sind. Man erhält die Abstandsmatrix A' :

$$(\pi_A, A, P_A) \mapsto (\pi_A|_{\{v\}}, A', P_A) \text{ mit } A' = (a'_{i,j})_{x+1, x+1} \text{ und } a'_{i,j} = \begin{cases} a_{i,j} \in A & i \neq v \wedge j \neq v \\ 0 & i = v \wedge j = v \\ \infty & \text{sonst.} \end{cases}$$

Damit A' weiterhin eine Blockdiagonalmatrix ist, soll v die letzte Zeile/Spalte von A' bilden. Ist $v = s$ oder $v = t$, so gilt der Einzelblock v als markiert.

3.4.2 Deaktivierung eines Knotens v

Prinzipiell wird hierbei aus der Abstandsmatrix jedes Zustands die v -Zeile/Spalte entfernt. Bei $v = s$ oder t hingegen wandelt man s bzw. t in s' bzw. t' um (siehe Abbildung 3.17). Es ist jedoch zu beachten, dass diese Deaktivierung wie beim gewöhnlichen all-terminalen Zusammenhang in verschiedene Fälle unterteilt werden muss. Durch die Markierung ergeben sich zudem weitere Unterfälle:

1. $\{v\}$ ist ein Einzelblock in π_A und es existieren weitere Blöcke.

Dabei gelten auf Grund der Markierungen folgende Varianten als Einzelblock:

- (A) $\{v\}$
- (B) $\{v, s'\}$
- (C) $\{v, t'\}$
- (D) $\{v, s', t'\}$

- (a) $v \neq s, t$

Bei (A) bildet sich eine Komponente, die s, t nicht enthält, der s, t -Zusammenhang ist damit weiterhin möglich. Der Zustand wird nicht entfernt.

Bei (B) und (C) bildet sich eine Komponente, die s oder t enthält. Der s, t -Zusammenhang ist damit nicht mehr erreichbar. Der Zustand wird gelöscht.

Bei (D) bildet sich eine Komponente, die s und t enthält. Ist $a_{s', t'} > D$, so kann kein D -Weg mehr erzeugt werden. Der Zustand ist zu löschen. Andernfalls existiert bei $a_{s', t'} \leq D$ ein D -Weg, womit s' und t' über einen D -Weg verbunden sind. Ist zusätzlich die letzte Fusion bereits ausgeführt worden, so sind die Voraussetzungen für eine vorzeitige Entfernung erfüllt (siehe Kapitel 2.3). Diese kann somit ausgeführt werden. Ist eine der Voraussetzungen (D -Weg und letzte Fusion) unerfüllt, so wird der Zustand nicht entfernt.

- (b) $v = s$ oder $v = t$

Bei (A) siehe Punkt 1 (a)(B) bzw. Punkt 1 (a)(C).

Bei (B) und (C) siehe Punkt 1 (a)(D).

Ein Fall (D) lässt sich nicht konstruieren, da s, t nach der dritten Bedingung

der Baumzerlegung nur einmal deaktiviert werden können und es im Fall (D) bereits sind.

2. Der Einzelblock $\{v\}$ ist der einzige Block in π_A .

Dabei gelten auf Grund der Markierungen folgende Varianten als Einzelblock:

- (A) $\{v, s'\}$
- (B) $\{v, t'\}$
- (C) $\{v, s', t'\}$

Mit der Deaktivierung von v entstünde ein leerer Block. Dies tritt nach Definition der Baumzerlegung erst ein, wenn wir die Wurzel erreicht haben und somit am Ende der Berechnung sind. Eine Variante $\{v\}$ kann folglich nicht auftreten, da laut Definition am Berechnungsende s und t in der Partition enthalten sein müssen.

(a) $v \neq s, t$

(A) und (B) können nach oben stehender Begründung nicht auftreten. Bei (C) bildet sich eine Komponente, die s und t enthält. Weitere Grundschritte werden nicht mehr folgen. Ist $a_{s',t'} > D$, so existiert kein D -Weg. Der Zustand wird gelöscht. Andernfalls existiert bei $a_{s',t'} \leq D$ ein D -Weg. Der Zustand wird entfernt und sein Wahrscheinlichkeitspolynom zum DC2RP addiert.

(b) $v = s$ oder $v = t$

Bei (A) und (B) siehe Punkt 2 (a)(C).

(C) kann nicht eintreten, da s, t bereits deaktiviert sind.

3. Ansonsten gilt für die restlichen Zustände:

(a) $v \neq s, t$

$$(\pi_A, A, P_A) \mapsto (\pi_A - v, A', P_A)$$

$$\text{mit } A' = (a'_{i,j})_{x-1, x-1} \text{ mit } a'_{i,j} = (a_{i,j} \in A \mid i \neq v \wedge j \neq v)$$

(b) $v = s$ oder $v = t$

Der Index s bzw. t in A wird zu s' bzw. t' . Analog in der Partition.

Zusammenfassung:

Nach den Punkten 1, 2 und 3 können Zustände mit identischen Abstandsmatrizen entstanden sein (die Partitionen stimmen dadurch automatisch überein). Diese sind in einem Zustand zusammenzufassen, indem die Partition und die Abstandsmatrix übernommen werden und als Wahrscheinlichkeitspolynom die Summe der Wahrscheinlichkeitspolynome der Zustände mit identischer Partition π_A und Matrix A eingesetzt wird.

3.4.3 Untersuchung einer Kante $e = \{u, v\}$

Dieser Schritt ist eine der Schlüsselstellen, um die in Kapitel 3.2 aufgeworfenen Probleme zu lösen. Er ist daher ausführlicher dargelegt als die beiden obigen Schritte.

Wie in Kapitel 2 wird e auf Ausfall und Intaktheit hin untersucht. Dazu duplizieren wir wie gewohnt alle Zustände. Auf das erste Duplikat wendet man den Kantenausfall an und auf das zweite Duplikat die Intaktheit.

Ausfall der Kante e

Das Polynom des Zustands wird mit der Wahrscheinlichkeit für den Ausfall $(1 - p)$ multipliziert. Da mit dem Ausfall kein Teilgraph verändert wird, ist auch an der Partition und der Abstandsmatrix keine Veränderung vorzunehmen. Man erhält somit den Zustand

$$(\pi_A, A, (1 - p) P_A)$$

Intaktheit der Kante e

Das Polynom des Zustands wird mit der Wahrscheinlichkeit für die Intaktheit p multipliziert. Mit der Intaktheit von e sind u und v verbunden und besitzen den Abstand 1:

$$a_{u,v} = a_{v,u} = 1$$

Wie in Kapitel 3.2.4 beschrieben, können neue kürzeste Wege entstanden sein. Im Gegensatz zur Idee in Kapitel 3.2 ist es nun aber möglich, diese Wege zu erfassen. Um unnötigen Aufwand zu vermeiden, unterscheiden wir die Lage von u und v in folgenden zwei Fällen:

(a) u und v befinden sich in verschiedenen Blöcken:

Interpretation für die durch den Zustand dargestellten Teilgraphen: Zwei Komponenten eines Teilgraphen wurden mit einer Kante verbunden. Die Kanten der Komponenten selbst bleiben dadurch unverändert erhalten. Folglich ändern sich die Abstände innerhalb der Komponenten und damit innerhalb der Blöcke nicht (siehe Abbildung 3.18).

Sei U als Menge aller Knoten definiert, die sich im selben Block wie u befinden (einschließlich u). Analog definiere man V für den Knoten v . Damit sind nur die Abstände zwischen den Knoten aus U und V zu ermitteln. Die Abstände aller Knoten in U (speziell zu u) und in V (speziell zu v) sind bereits bekannt. Durch e kennen wir den Abstand von u zu v . Die gesuchten Abstände lassen sich somit wie folgt bestimmen:

$$\forall i \in U \forall j \in V : a_{i,j} = a_{j,i} = a_{i,u} + a_{u,v} + a_{v,j} \text{ mit } a_{u,v} = 1 \quad (3.2)$$

Befinden sich s und/oder t bzw. s', t' in den Blöcken, wird durch (3.2) automatisch der Abstand der neu hinzugefügten Knoten zu den Terminalknoten erzeugt.

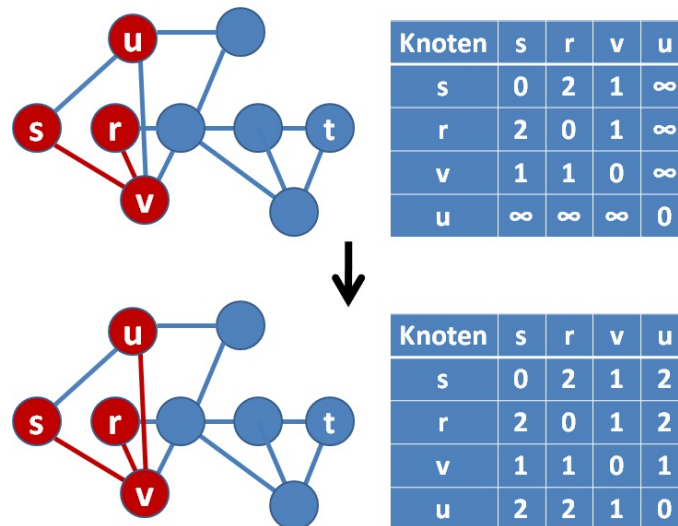


Abbildung 3.18: Umsetzung für u, v in verschiedenen Blöcken

(b) u und v befinden sich im selben Block:

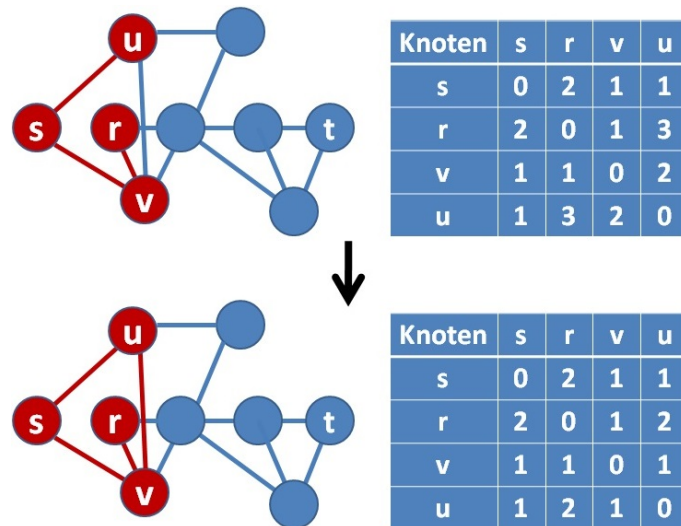
Es müssen keine Blöcke vereinigt werden. Wir haben lediglich eine Aktualisierung der Abstände im Block vorzunehmen (siehe Abbildung 3.19). Dafür dient folgende Überlegung als Grundlage:

Entsteht durch das Einfügen von e ein neuer kürzester Weg zwischen zwei Blockknoten x und y , so muss dieser über e führen. Täte er dies nicht, so wäre sein Verlauf und seine Länge bereits erfasst und er wäre kein neuer kürzester Weg. Damit ist es möglich, durch $d(x, u) + d(u, v) + d(v, y)$ die Länge eines Weges von x nach y über e zu bestimmen und durch den Vergleich mit $d(x, y)$ zu entscheiden, ob ein neuer kürzester Weg gefunden wurde. Wenn ja, so ist der neue Wert einzutragen, andernfalls ist keine Änderung vorzunehmen.

Es ist zu beachten, dass im Allgemeinen

$$l_1 = d(x, u) + d(u, v) + d(v, y) \neq d(x, v) + d(u, v) + d(u, y) = l_2$$

gilt. Man muss also beide Werte berechnen und den kleineren mit $d(x, y)$ vergleichen. Natürlich können dabei auch $d(x, u), d(x, v), d(y, u), d(y, v)$ neue Werte annehmen.


 Abbildung 3.19: Umsetzung für u, v im selben Block

Behauptung:

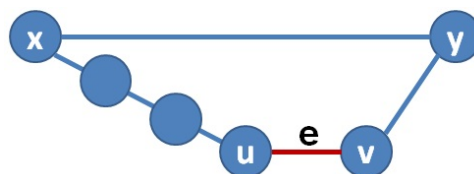
Für alle Blockknoten x, y kann der obige Vergleich angewandt werden, ohne zuerst die neuen Werte $d(x, u), d(x, v), d(y, u), d(y, v)$ ermitteln zu müssen.

Beweis:

Wird der bisherige Matrixeintrag $d(x, u)$ durch Einfügen von e zu einem falschen Wert, folgt, dass der bisherige Eintrag $d(x, v)$ weiterhin korrekt ist. Denn aus der Falschheit des alten Wertes $d(x, u)$ (dessen zugehöriger kürzester Weg nicht über e führt) folgt, dass der neue kürzeste Weg von x nach u nun über e führt und dabei zuerst v erreicht. Ansonsten wäre der bisherige Wert $d(x, u)$ weiterhin korrekt gewesen. Dadurch führt der kürzeste Weg von x nach v nicht über e , womit sich $d(x, v)$ nicht ändert und damit wahr bleibt. Analog gilt: $d(x, v)$ falsch $\rightarrow d(x, u)$ richtig. Selbige Überlegungen sind bei $d(y, u)$ und $d(y, v)$ anzuwenden. Folglich können maximal zwei Abstände zur gleichen Zeit falsch sein.

↪

- a) Erzeugen sowohl l_1 als auch l_2 einen falschen Abstandswert, so muss in jeder der beiden Summen der richtige der beiden Abstände unabhängig von e sein und der kürzeste Weg des falschen Abstandes muss über e verlaufen.


 Abbildung 3.20: $d(x, u)$ und $d(y, u)$ falsch, damit sind l_1, l_2 falsch

Wie aus Abbildung 3.20 zu entnehmen ist, führt der kürzeste Weg von x nach y nicht über e . Es ist also kein neuer kürzester Weg entstanden und der Vergleich von l_1, l_2 mit dem alten Wert $d(x, y)$ liefert das richtige Ergebnis $d(x, y)$.

- b) In allen anderen Fällen ist wenigstens eine der beiden Summen l_1, l_2 richtig, so dass der Vergleich mit $d(x, y)$ einen korrekten Abstandswert erzeugt.

□

Bezeichne W die Menge aller Knoten, die im selben Block wie u, v liegen (einschließlich u und v). So ergibt sich folgende Formel für den Vergleich:

$$\forall x, y \in W : a_{x,y}^{\text{neu}} = \min(a_{x,y}, d(x, u) + d(u, v) + d(v, y), d(x, v) + d(u, v) + d(u, y)) \quad (3.3)$$

Bemerkung 3.4.1

Befinden sich s und/oder t bzw. s', t' im Block, wird durch (3.3) automatisch der Abstand der Knoten zu den Terminalknoten aktualisiert.

Im Gegensatz zur Baumzerlegung mit Partitionen können hier unterschiedliche Zustände entstehen, wenn sich u und v im selben Block befinden. Die Kantenuntersuchung ist also bei allen Zuständen auszuführen.

Bemerkung 3.4.2

Es ist zu beachten, dass im Falle eines gewichteten Graphen die Matrixeintragung für eine gewichtete Kante $\{u, v\}$ nicht 1 ist sondern dem Kantengewicht entspricht. Existiert bei intakter Kante bereits ein kleinerer Eintrag, so ist das Kantengewicht demzufolge nicht zu übernehmen. Andernfalls kann die Kantenuntersuchung unter Beachtung des Kantengewichts analog zum ungewichteten Fall durchgeführt werden.

Bemerkung 3.4.3

Die Zusammenfassung der Zustände erfolgt analog zur Deaktivierung.

3.4.4 Fusion zweier Teilbäume t_1, t_2

Die Fusion wurde bereits in Kapitel 2 ausführlich erläutert, so dass lediglich die Verwendung der Abstandsmatrix in das Verfahren integriert werden muss.

Dazu stellt man erneut jeden Zustand der Zustandsmenge von t_1 jedem Zustand der Zustandsmenge von t_2 gegenüber. Aus jedem Zustandspaar erhält man einen neuen Zustand, indem ihre Wahrscheinlichkeitspolynome multipliziert, und die Partitionen und Abstandsmatrizen fusioniert werden. Die Fusion der Partitionen erfolgt auf bekannte Art und Weise.

Fusion der Abstandsmatrizen

Für die Fusion wähle man die Matrix A aus t_1 und integriere die Matrix B aus t_2 in ihr. Dabei kann auf Inhalte der vorherigen drei Grundschriffe zurückgegriffen werden. Eine Manipulation der Wahrscheinlichkeiten findet jedoch nicht mehr statt; dies erfolgte bereits durch die Multiplikation der Polynome. Die Fusion zweier Zustände lässt sich als Vereinigung ihrer Teilgraphen interpretieren (siehe Abbildung 3.21). Damit erhalten die durch A dargestellten Teilgraphen neue Kanten und neue kürzeste Wege. Um bei der Verarbeitung ein geordnetes Vorgehen zu ermöglichen, betrachte man neue Kanten und neue kürzeste Wege getrennt.

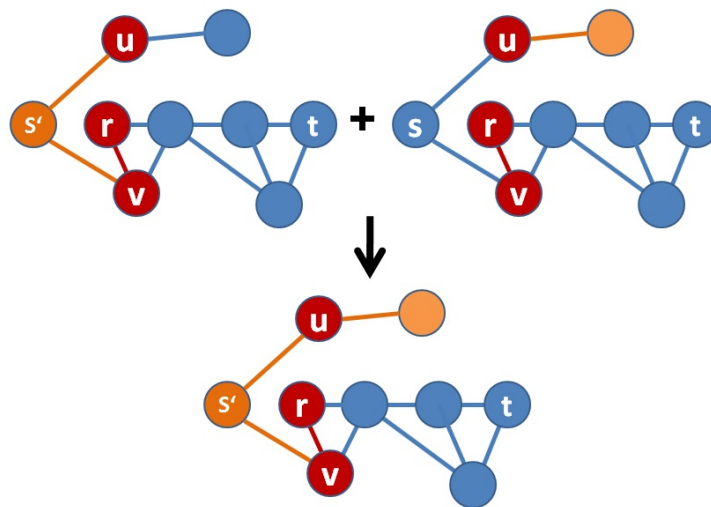


Abbildung 3.21: Interpretation der Fusion als Vereinigung zweier Teilgraphen

Die Integration erfolgt in drei Stufen:

Stufe 1: Übernahme fremder Knoten

A und B enthalten prinzipiell dieselben Knoten, da sie durch den Join dieselbe aktive Menge X_i besitzen. Mittels Markierung durch s, t bzw. s', t' ist es möglich, dass A bzw. B s' und/oder t' enthält (aber nicht s und t getrennt, sonst wären sie in der aktiven Menge und damit in beiden Matrizen enthalten). Befindet sich s' und/oder t' in B , so fügt man den/die Knoten in A mittels Knotenaktivierung ein.

Stufe 2: Übernahme von Kanten

Alle Kanten aus B werden nacheinander mittels Kantenuntersuchung intakter Kanten schrittweise übernommen.

Zwei Blöcke werden vereinigt, wenn sie je einen Endknoten einer Kante enthalten. Der Arbeitsaufwand für solche Kanten entspricht der Größe der beiden Blöcke. Der Arbeitsaufwand für Kanten mit Endknoten desselben Blocks entspricht der Blockgröße.

Daher ist es günstig, Kanten der ersten Art so spät wie möglich zu übernehmen. Dadurch ist der zu bearbeitende Block für Kanten der zweiten Art wesentlich kleiner und der Aufwand wird reduziert.

Stufe 3: Übernahme kürzester Wege

Ein Eintrag $b_{u,v}$ aus B ist zu übernehmen, wenn er kleiner als der Eintrag $a_{u,v}$ in A ist (sonst würde man damit den Wert eines längeren Weges übernehmen). $b_{u,v}$ steht für die Länge eines kürzesten Weges f von u nach v . Angenommen wir ersetzen die Begriffe „die Kante $e = \{u, v\}$ “ durch „der kürzeste Weg f von u nach v “ in Kapitel 3.4.3 und durchdenken die dadurch entstandenen Überlegungen, so ist leicht einzusehen, dass die Aussagen der Kantenuntersuchung auch für Längen $b_{u,v}$ kürzester Wege gelten. Damit erfolgt die Übernahme von $b_{u,v}$ analog zur Übernahme von Kanten.

Zusammenfassung:

Nach der Fusion können Zustände mit identischen Abstandsmatrizen entstanden sein. Ihre Zusammenfassung erfolgt analog zur Deaktivierung.

3.4.5 Behandlung der Knoten s und t

Die hier verwendete Art der Markierung erzeugt verschiedene Arten des Auftretens von s und t . Es soll an dieser Stelle ein abschließender Blick auf ihre Kompatibilität mit den Grundschritten geworfen werden:

- *Weder s noch t wurde aktiviert.*
Dadurch entfallen einige Spezialfälle in den Grundschritten. Die Blockknoten erhalten dennoch ihre Abstände untereinander.
- *Ein Terminalknoten ist aktiv, der andere wurde noch nicht aktiviert.*
Die Aktivierung des ersten Terminalknotens erfolgt wie die Aktivierung der übrigen Knoten. Unter Beachtung der Spezialfälle der Deaktivierung erfolgt die Anwendung auf diesen aktiven Knoten analog zu den übrigen Knoten.
- *Ein Terminalknoten ist deaktiviert, der andere wurde noch nicht aktiviert.*
Die Deaktivierung erfolgt unter Anwendung der beschriebenen Spezialfälle. Danach wird der deaktivierte Knoten wie ein aktiver Knoten weitergeführt. Durch die Weiterführung als aktiver Knoten wird der Terminalknoten in den restlichen Grundschritten wie die übrigen aktiven Knoten behandelt.
- *s und t sind aktiv.*
Beide Terminalknoten werden als aktiv geführt. Die Grundschriffe arbeiten damit regulär. Sobald sich s und t im selben Block befinden, kann jederzeit eine vorzeitige Entfernung vorgenommen werden, vorausgesetzt, die Bedingungen dafür sind

erfüllt (siehe unten). Die Vereinigung des s - und t -Blocks während der Kantenuntersuchung erzeugt keine Probleme, da s und t aktiv sind.

- *Ein Terminalknoten ist aktiv, der andere deaktiviert.*
Die Deaktivierung erfolgt durch Anwendung der Spezialfälle. Der deaktivierte Knoten wird für die Kantenuntersuchung, Aktivierung und Fusion als aktiv angesehen. Damit arbeiten die Schritte weiterhin korrekt.
- *s und t sind deaktiviert.*
Die Spezialfälle der Deaktivierung müssen für den entsprechenden Block beachtet werden. Für die restlichen drei Schritte sind die Knoten aktiv, daher arbeiten sie korrekt.

3.4.6 Vorzeitiges Entfernen von Zuständen

Die in Kapitel 2.3 beschriebene vorzeitige Entfernung von Zuständen kann auch in diesem Fall angewandt werden. Zusätzlich zu den Bedingungen, dass die letzte Fusion durchgeführt worden ist und s, t bzw. s', t' im selben Block zu liegen haben, muss notwendigerweise gelten: $a_{s,t} \leq D$ (entsprechend auch für Kombinationen mit s' und t').

3.4.7 Vorzeitige Eliminierung von Zuständen durch Abstandsanalyse

Das vorzeitige Entfernen von Zuständen erlaubt es uns, Zustände aus der Polynom-berechnung herauszunehmen. Diese haben einen gewissen Punkt erreicht, an dem sie, unabhängig von der Entwicklung der weiteren Ereignisse, stets zum s, t -Zusammenhang führen.

Im Gegensatz dazu ist die vorzeitige Eliminierung von Zuständen denjenigen Zuständen angedacht, die auf Grund ihrer Struktur auch bei Intaktheit aller verbleibenden, nicht untersuchten Kanten nicht mehr in der Lage sind, einen D -Weg zu generieren. Einige dieser Zustände werden bereits durch die Deaktivierung entfernt, nämlich genau dann, wenn eine Komponente mit nur einem Terminalknoten oder beiden Terminalknoten und $a_{s',t'} > D$ entsteht.

Sei beispielsweise s bereits aktiviert worden und t noch nicht. Ein Abstand $d(s, t)$ ist zu diesem Zeitpunkt noch nicht in A erzeugt. Dennoch kann man sich leicht einen Graphen und dazu einen Teilgraphen überlegen, bei denen ein kürzester s, t -Weg über untersuchte und nicht untersuchte Kanten verläuft, so dass seine Länge größer als D ist. Wenn man ein geeignetes Berechnungsverfahren verwendet, das den Abstand $d(s, t)$ im Teilgraphen und Restgraphen in günstiger Zeit liefert, so können solche Zustände frühzeitig eliminiert werden.

Ansatz

Für die Entfernungsberechnung ist eine einfache Breitensuche nicht ausreichend. Das Problem besteht einerseits darin, dass durch den Einsatz von Reduktionen gewichtete Kanten entstanden sein können. Andererseits ist zwar die Gestalt des nicht untersuchten Teils von G bekannt, aber nicht die der Teilgraphen. Um nicht jeden Teilgraphen speichern zu müssen, wurden die Abstandsmatrizen eingeführt.

Es ist leicht einzusehen, dass ein Weg über untersuchte und nicht untersuchte Kanten die untersuchten Kanten auf zweierlei Arten nutzt:

1. *Zwischen zwei nicht untersuchten Kanten des Weges befindet sich eine gewisse Anzahl an untersuchten Kanten.*

Etwas lax formuliert lässt sich dieser Sachverhalt so interpretieren: Der Weg geht in einen Block hinein und kommt aus ihm wieder heraus (aus einem anderen Block kann er auf Grund der fehlenden Verbindung nicht herauskommen). Da wir alle nicht untersuchten Kanten kennen, ist damit bekannt, bei welchem Blockknoten u der Weg hinein und bei welchem v er wieder herauskommt. Da die Länge eines kürzesten Weges als $a_{u,v}$ gespeichert wurde, müssen wir die intakten untersuchten Kanten nicht kennen. Ein solches Wegstück kann durch eine Einzelkante mit dem Wert $a_{u,v}$ ersetzt werden.

2. *Vor oder nach einer nicht untersuchten Kante befinden sich nur noch untersuchte Kanten.*

Wiederum etwas lax interpretiert, bedeutet der Sachverhalt folgendes: Der Weg geht in einen Block hinein und kommt nicht wieder heraus. Damit endet der Weg im Block. Und zwar an einem Terminalknoten. Wir kennen wieder den Eintrittsknoten u und den Endknoten (o.B.d.A.) s . Durch die Matrix ist $d(u, s) = a_{u,s}$ bekannt. Ein solches Wegstück kann durch eine Einzelkante mit dem Wert $a_{u,s}$ ersetzt werden.

Der Graph der Abstandsanalyse

Ist eine Abstandsanalyse für einen Zustand durchzuführen, muss zuerst der Graph definiert werden, auf den die Analyse angewandt wird. Wie oben zu sehen ist, sind alle nicht untersuchten Kanten bekannt (Menge $E' \subseteq E$), ebenso wie die Abstände zwischen den Knoten der aktiven Menge X_i . Um die kürzesten Wege zwischen den aktiven Knoten darzustellen, füge man zwischen allen aktiven Knoten u, v eine Kante $\{u, v\}$ mit Kantengewicht $a_{u,v}$ ein, wenn $a_{u,v} \notin \{\infty, 0\}$. Dann bilden alle Knoten eines Blocks eine Clique. Damit erhält man als **Analysegraphen** (wie in Abbildung 3.22 zu sehen) einen schlichten, gewichteten Graphen $G'' = (V'', E'')$ mit ausgezeichneten Knoten s und t und

$$V'' = V' \cup X_i, E'' = E' \cup \{\{u, v\} \mid (a_{u,v} \notin \{\infty, 0\}) \in A\}$$

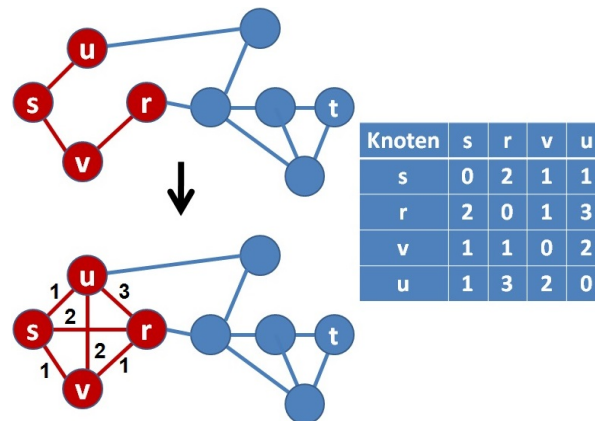


Abbildung 3.22: Graph der Abstandsanalyse

Bemerkung 3.4.4

In der Zuverlässigkeitsberechnung wird dieser Graph nicht erzeugt. Er dient jedoch als Modell, um zu erklären, wie die entstehenden Informationen in der Analyse verarbeitet werden.

Die Abstandsanalyse

Zur Berechnung von $d(s, t)$ in unserem Analysegraphen G'' verwenden wir den Dijkstra-Algorithmus [Dij59].

Einen kompakten Überblick mit den Beweisen zur Korrektheit und dem Nachweis der Laufzeit bietet [Internet01]. Die vorliegende Funktionsweise wurde aus [Internet02] entnommen:

Die Grundidee des Algorithmus ist, ab einem Startknoten die kürzest möglichen Wege weiter zu verfolgen und längere Wege beim Updaten auszuschließen. Er besteht im wesentlichen aus diesen Schritten:

1. *Initialisiere alle Knoten mit der Distanz unendlich; die des Startknotens mit 0.*
2. *Markiere die Distanz des Startknotens als permanent, alle anderen Distanzen als temporär.*
3. *Setze Startknoten als aktiv.*
4. *Berechne die temporären Distanzen aller Nachbarknoten des aktiven Knotens durch Addition von dessen Distanz mit den Kantengewichten.*
5. *Wenn eine solche berechnete Distanz für einen Knoten kleiner ist als die aktuelle, aktualisiere die Distanz und setze den aktuellen Knoten als Vorgänger. Dieser Schritt wird auch als Update bezeichnet und bildet die zentrale Idee des Dijkstra-*

Algorithmus.

6. *Wähle einen Knoten mit minimaler temporärer Distanz als aktiv. Markiere seine Distanz als permanent.*
7. *Wiederhole die Schritte 4-7, bis es keinen Knoten mit permanenter Distanz gibt, dessen Nachbarn noch temporäre Distanzen haben.*

Dieses Vorgehen gewährleistet, dass bei Erreichen eines Knotens kein kürzerer Pfad zu ihm existieren kann. Der Dijkstra-Algorithmus ist den Greedy-Algorithmen zuzuordnen, die in jedem Schritt die momentan aussichtsreichste Teillösung bevorzugen. Er liefert stets die optimale Lösung. Angenommen, der gefundene Weg ist nicht der kürzeste, so hätte man auch dessen kürzere Teilstrecke früher untersuchen müssen. Dann hätte man aber über die kürzere Teilstrecke den Zielknoten früher gefunden als auf dem längeren Weg. Mittels der Datenstruktur Heap erhält man eine Laufzeit von $O(n \log n + m)$.

Lage von s und t

Die Abstandsanalyse des Graphen G'' ist unabhängig von der Lage von s und t . Ob nun beide Terminalknoten noch nicht aktiviert wurden oder nur einer oder beide schon aktiviert sind, ist unerheblich. Im ersten Fall ist es möglich, dass ein kürzester Weg auch über untersuchte Kanten verläuft, im zweiten Fall ganz sicher und im dritten Fall ist es möglich, dass er auch über nicht untersuchte Kanten verläuft. Für jede mögliche Lage von s und t lässt sich ein Analysegraph G'' erzeugen. Damit ist die Abstandsanalyse stets anwendbar und liefert ein korrektes Ergebnis.

Anwendung

Prinzipiell liegt es im Ermessen des Benutzers, wann die Abstandsanalyse angewandt wird. Es ist allerdings zu beachten, dass damit für einen einzelnen Zustand ein Aufwand von $O(n \log n + m)$ entsteht. Von einer allzu häufigen Anwendung ist demzufolge abzuraten. Es können jedoch Richtlinien aufgestellt werden, um den Aufwand der Analyseanwendung zu verringern:

- In den frühen Phasen der Polynomberechnung sollte die Analyse seltener angewandt werden. Es sind noch viele Kanten nicht untersucht worden. Da diese während der Analyse als intakt angesehen werden, ist nicht mit zu eliminierenden Graphen zu rechnen. Im späteren Verlauf ist der Analysegraph kleiner und es ist zu erwarten, dass viele Kanten ausgefallen sind.
- Bei Aktivierungen werden keine Kanten verändert. Nach diesem Schritt ist eine Analyse unnötig.
- Nach einer Kantenuntersuchung entstehen meist viele Zustände, eine Analyse

bietet sich damit an. Jedoch zieht eine Aktivierung meist mehrere Kantenuntersuchungen mit sich. Dabei wird nur je eine Kante untersucht. Die Analyse sollte erst nach der letzten Untersuchung durchgeführt werden.

- Nach einer Fusion entstehen meist viele Zustände. Eine Analyse bietet sich damit an. Allerdings erst nach dem Zusammenfassen, um zusätzlichen Aufwand zu vermeiden.
- Folgt nach einer Fusion oder Kantenuntersuchung eine oder mehrere Deaktivierungen, so sollte die Analyse erst danach durchgeführt werden. Eine Deaktivierung ist im Vergleich zu den anderen Grundschritten schnell und danach sind Blockknoten entfernt. Damit hat man kleinere Cliques im Analysegraph. Der Rechenaufwand der Analyse ist damit nach den Deaktivierungen geringer, als davor.
- Ist D sehr groß und $d(s, t)$ nach einer Analyse durchschnittlich klein, so ist eine erneute Analyse später anzuwenden als bei großem $d(s, t)$ und/oder kleinem D .

3.5 Komplexität der beschränkten Zuverlässigkeitsberechnung

Die Komplexität des Algorithmus kann mittels der Verwendung der vier Grundschritte abgeschätzt werden. Sei I die Indexmenge der Baumknoten. Es ist bekannt, dass sich zwei adjazente Baumknoten X_i, X_j mit $i \neq j$ maximal um je einen Graphenknoten unterscheiden. Damit gibt es maximal $2|V|$ Elemente im Baum T und deshalb $|V|$ Knoten mit zwei oder mehr Kindern.

3.5.1 Speicheraufwand in Schritt i

In der zu Grunde liegenden Berechnung der s, t -Zuverlässigkeit ohne Weglängenbeschränkung beinhalten die Zustände Partitionen der aktiven Menge als Speicher der erzeugten Teilgraphen. Mit der Weite w der Baumzerlegung wird die Mächtigkeit der aktiven Menge nach oben beschränkt. Damit existiert ebenfalls eine obere Schranke für die Zahl der Zustände in einem Schritt i : Die Bellzahl $B(w)$ als Anzahl an Partitionen, die aus einer Menge der Mächtigkeit w gebildet werden kann (vergleiche dazu Kapitel 2.5).

Um eine Berechnung mit Weglängenbeschränkung zu gewährleisten, benötigt man zusätzliche Informationen. Dazu wurde die Abstandsmatrix A eingeführt. Sie enthält die Abstände der Knoten der aktiven Menge und zu s', t' , falls die Terminalknoten schon deaktiviert wurden. Da A quadratisch ist, sind in ihr maximal

$$(\max_{i \in I} |X_i \cup \{s, t\}|)^2 \leq (w + 3)^2$$

Elemente enthalten.

Die Anzahl der Zustände entspricht der Anzahl der verschiedenen erstellbaren Abstandsmatrizen. Die Anzahl der Matrizen ist gleich der Anzahl der verschiedenen Einträge. Damit wird die Anzahl der Möglichkeiten für einen einzelnen Eintrag durch die Länge $l \leq n - 1$, $n = |V|$ eines längsten Weges in G nach oben beschränkt. Zusätzlich gibt es die zwei Einträge 0 und unendlich.

Stellt man A als Blockdiagonalmatrix dar, so sind die Einträge der Hauptdiagonalen 0 (da $d(v, v) = 0$). Zudem gilt $d(u, v) = d(v, u) \forall u, v \in X_i$, wodurch wir nicht $(w + 3)^2$, sondern nur $f(w) = \frac{(w+2)(w+3)}{2}$ veränderliche Elemente (Anzahl der unteren Elemente einer unteren Dreiecksmatrix, nach Gauß) zählen. Die maximale Anzahl an Abstandsmatrizen (und damit an Zuständen) pro Iterationsschritt beträgt folglich

$$(n+1)^{\frac{(w+2)(w+3)}{2}} = (n+1)^{f(w)} \quad (3.4)$$

3.5.2 Komplexität der vier Grundschritte

Aktivierung

Jede Matrix wird um dieselbe Zeile und Spalte erweitert (siehe dazu Kapitel 3.4.1). Sie erhält damit maximal $2(w+2) + 1 = 2w + 5$ zusätzliche Einträge ($w+2$, da das $w+3$ -te Element erst jetzt eingefügt werden kann). Mit (3.4) ergeben sich damit für eine Aktivierung $(2w+5)(n+1)^{f(w)}$ Operationsschritte. Da $|V|$ viele Aktivierungen erfolgen, ergeben sich im Laufe der Berechnung insgesamt

$$|V|(2w+5)(n+1)^{f(w)} \in O\left(wn^{f(w)+1}\right)$$

Operationsschritte.

Deaktivierung

Nach Kapitel 3.4.2 wird jede Matrix um dieselbe Zeile und Spalte vermindert, wenn keine Spezialfälle vorliegen (deren Verarbeitung wäre maximal mit einer konstanten Anzahl $c \in \mathbb{N}$ Operationsschritten anzusetzen). Es ergeben sich $2(w+3) - 1$ Eliminierungen und c Schritte für die Spezialfälle. Da alle Knoten wieder deaktiviert werden müssen, erhält man analog zu Aktivierung

$$|V|(2w+5+c)(n+1)^{f(w)} \in O\left(wn^{f(w)+1}\right)$$

Operationsschritte.

Kantenuntersuchung

Bei einer Kantenuntersuchung werden zwar die Zustände dupliziert, aber die Hälfte davon wird nicht bearbeitet, da der Kantenausfall nicht zur Aktualisierung der Matrix führt. Für die Komplexität werden die beiden Fälle für die Lage der Endknoten u, v der Kante e aus Kapitel 3.4.3 betrachtet:

- u, v liegen in verschiedenen Blöcken X, Y :

Dann befinden sich, unter Annahme des worst case, in jedem Block maximal $\left\lceil \frac{w+3}{2} \right\rceil$ Knoten. Da für jeden Knoten aus X eine Berechnung für jeden Knoten aus Y in Form von zwei Additionen und der Eintragung in die Matrix (linearer Aufwand $c \in \mathbb{N}$) durchgeführt wird, erhält man:

$$c \left\lceil \frac{w+3}{2} \right\rceil \left\lceil \frac{w+3}{2} \right\rceil \leq c \frac{(w+3)(w+3)}{4} \in O(w^2)$$

Operationsschritte für einen Zustand.

- u, v liegen im selben Block X :

Jeder der $w+3$ Knoten muss mit den $w+2$ restlichen Knoten bezüglich eines neuen Abstandes verglichen werden. Dazu sind 4 Additionen, ein Vergleich, zwei Eintragungen nötig. Diese werden mit $c \in \mathbb{N}$ abgeschätzt. Auf Grund der Symmetrie der Matrix ist nur die Hälfte der Vergleiche erforderlich. Man benötigt also

$$c \frac{(w+2)(w+3)}{2} \in O(w^2)$$

Operationsschritte.

Da für jede Matrix einer der beiden Fälle eintreten kann und nur $|E|$ Kanten zu untersuchen sind, werden im Laufe der Berechnung

$$O(w^2 n^{f(w)+2})$$

Operationsschritte benötigt.

Fusion

Wie in Kapitel 3.4.4 beschrieben, ist die Fusion lediglich eine Erweiterung der Kantenuntersuchung, die dasselbe Verfahren anwendet. Die Fusion zweier Zustände, bzw. ihrer Matrizen A und B , entspricht einer Mehrfachanwendung der Kantenuntersuchung. Im worst case wird für jeden Eintrag in B eine Kantenuntersuchung in A durchgeführt. Die Anzahl der Operationsschritte einer Fusion liegt damit in $O((w^2)^2) = O(w^4)$.

Da die Fusion durch die Kombination

$$\left((n+1)^{f(w)}\right)^2$$

Zustände erzeugt und sich maximal $|V|$ Joinknoten im Baum befinden, bei denen diese Berechnung anzuwenden ist, benötigt man insgesamt

$$|V| \left((n+1)^{f(w)}\right)^2 O(w^4) \in O\left(w^4 n^{f(w)^2+1}\right)$$

Operationsschritte.

Gesamtkomplexität

Mit Durchführung aller Operationsschritte erhält man damit für die Berechnung des DC2RP unter Verwendung einer Baumzerlegung eine Komplexität von

$$O\left(w n^{f(w)+1}\right) + O\left(w n^{f(w)+1}\right) + O\left(w^2 n^{f(w)+2}\right) + O\left(w^4 n^{f(w)^2+1}\right) \in O\left(w^4 n^{f(w)^2+1}\right)$$

Nur für $f(w) = 1$ ist der dritte Summand größer, dann besteht der Graph aus einem Knoten und $R_{s,t}(G, p, D) = 1$. Für alle anderen Graphen ist der vierte Summand größer. Das Verfahren ist damit für Graphen beschränkter Baumweite polynomial in der Anzahl der Knoten. Besonders stark wirkt sich dabei die Weite der verwendeten Baumzerlegung auf die Laufzeit aus. Bereits bei einer allgemein als klein angesehenen Weite von $w = 3$ liegt die Laufzeit in $O(n^{101})$. Zum Vergleich: Angenommen, man könne pro Millisekunde so viele Rechenschritte ausführen wie es Sandkörner an allen Stränden dieser Welt gibt, dann bräuchte man zur Berechnung von 2^{100} per $2 * 2 * 2 * 2 * \dots$ mehr Zeit, als seit dem Urknall vergangen ist. Die Berechnung des DC2RP nach dem hier vorgestellten Verfahren ist damit nicht effizient.

Allerdings ist zu beachten, dass sich die Komplexität aus der größten der obigen vier Summanden ergibt. Dieser steht für die Fusionen, von denen üblicherweise weniger ausgeführt werden müssen, als z.B.: Kantenuntersuchungen, die bei $w = 3$ mit $O(n^{12})$ für Graphen mit kleiner Knotenzahl immerhin effizient sind.

Zudem ist zu beachten, dass die Weglängenbeschränkung nur in der Abstandsanalyse auftritt, welche in den Grundschrinen und damit in der Ermittlung der Komplexität nicht auftritt. Für den worst case ist zwar anzunehmen, dass durch die Analyse kein Zustand entfernt wird, aber in der Praxis ist das vor allem bei kleinen Weglängenbeschränkungen (klein im Sinne von $D \approx d(s, t)$ in G) nicht der Fall, sodass das Verfahren bei kleiner Weite und kleiner Weglängenbeschränkung dennoch eine effiziente Laufzeit besitzen kann.

4 Auswertung und offene Probleme

In dieser Arbeit konnte die Weglängenbeschränkung in das Berechnungsverfahren des s, t -Zusammenhangspolynoms unter Verwendung einer Baumzerlegung integriert werden. Dafür wurde das Konzept der Zustandsspeicherung in Partitionen um Abstandsmatrizen erweitert. Man kann nun zusätzlich Informationen über kürzeste Wege zwischen den Knoten speichern. In Folge dessen wurde die Formulierung der vier Grundschriffe auf die Verwendung der Abstandsmatrizen angepasst. Es wurde ersichtlich, dass die Weglängenbeschränkung keinen Einsatz in der Anwendung der Grundschriffe findet. Zwischen den Schritten kann man die Beschränkung für die Elimination unzulässiger Zustände verwenden. Dafür wurde ein Modell etabliert, auf das der Dijkstra-Algorithmus anwendbar ist.

Zudem konnte gezeigt werden, dass die Laufzeit des Verfahrens für Graphen beschränkter Baumweite polynomial in der Anzahl der Graphenknoten ist, trotz erhöhter Informationsdichte. Auf Grund der ermittelten Komplexität ist das Verfahren jedoch als nicht effizient einzustufen. Mit den Abstandsmatrizen konnte die Struktur des alten Verfahrens größtenteils übernommen werden. Es ist weiterhin zu klären, ob nicht effizientere Berechnungsverfahren zu finden sind.

Damit ergeben sich mehrere Ansatzpunkte für die weitere Forschung:

1. Erstellung eines besseren Speicherkonzepts als die Abstandsmatrix.
2. Untersuchung einer effizienteren Einbindung der Weglängenbeschränkung in das Verfahren.
3. Entwicklung eines komplett neuen, auf einer Baumzerlegung basierenden Verfahrens zur Zuverlässigkeitsberechnung.
4. Weiterentwicklung des Verfahrens in eine Abschätzung der Zuverlässigkeit.

Literaturverzeichnis

- [Adl09] Isolde Adler, Baumzerlegungen, Algorithmen und Logik, Skript zum WS 2009/10, Goethe-Universität Frankfurt
- [Arn89] Stefan Arnborg, Andrzej Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, Vol. 23, No. 1., pp. 11-24, (April 1989)
- [Bal79] M. O. Ball. Computing network reliability, *Operations Research*. Vol. 27, pp. 823–836, (1979)
- [Bal80] M. O. Ball. Complexity of network reliability computation. *Networks*, 10, 153-165, (1980)
- [Bal83] J.–S. Provan and M.–O. Ball, The complexity of counting cuts and of computing the probability that a graph is connected, *SIAM Journal on Computing*, Vol. 12, pp. 777–788, (1983)
- [Bal86] M. O. Ball. Computational complexity of network reliability analysis: An overview. *IEEE Transactions on Reliability*, R-35(3):230-239, (1986)
- [Bet04] Nadja Betzler, Rolf Niedermeier, and Johannes Uhlmann. Tree Decompositions of Graphs: Saving Memory in Dynamic Programming. *CTW*, page 56-60, (2004)
- [Bod95] Hans L. Bodlaender, John R. Gilbert, Hjálmtýr Hafsteinsson, und Ton Kloks. Approximating Treewidth, Pathwidth, Frontsize, and Shortest Elimination Tree. *J. Algorithms*, 18(2):238-255, (1995)
- [Bod97] Hans L. Bodlaender. Treewidth: Algorithmic techniques and results. *Proceedings 22nd International Symposium on Mathematical Foundations of Computer Science, MFCS'97, Lecture Notes in Computer Science*, volume 1295, Igor Privara and Peter Ruzicka (editors), Springer-Verlag, Berlin, 29-36, (1997)
- [Bod98] Hans L. Bodlaender. A Partial k-Arboretum of Graphs with Bounded Treewidth. *TCS* 209(1-2), 1-45, (1998)
- [Bru58] N. G. de Bruijn. *Asymptotic methods in analysis*. New York:Dover,npages 102–109, (1958)
- [Can98] H. Cancela, and M. El Khadiri. Series-parallel reductions in Monte Carlo network reliability evaluation. *IEEE Transactions on Reliability*, 47(2):159-164, (1998)

- [Can01] H. Cancela and L. Petingi. Diameter constrained network reliability: exact evaluation by factorization and bounds. Proceedings of the International Conference on Industrial Logistics, Okinawa, Japan, pp. 359–366, (2001)
- [Can03] P. Burgos, H. Cancela, A. Godoy and L. Petingi. Source-terminal reliability maximization with path length constraints. International Conference on Industrial Logistics (ICIL'03), pp. 76-83, Vaasa, Finland, (2003)
- [Can04] H. Cancela and L. Petingi. Reliability of Communication Networks with Delay Constraints: Computational Complexity and Complete Topologies. International Journal of Mathematics and Mathematical Sciences, Vol. 2004, No. 29, pp 1551–1562, (2004)
- [Can06] H. Cancela and L. Petingi. On the Characterization of the Domination of a Diameter Constraint Network Reliability Model. Discrete Applied Mathematics, Vol. 154, No. 13, pp. 1885–1896, (2006)
- [Can07] H. Cancela and L. Petingi. Properties of a generalized Source-to-All-terminal network reliability model with diameter constraints. Omega (The International Journal of Management Science) 35(6), pp. 659-670, (2007)
- [Can10] H. Cancela, M. El Khadiri and L. Petingi. An exact Method to evaluate the Source-to-Terminal Diameter Constrained Reliability of a Communication Network. Proceedings of the International Conference on Industrial Logistics, Rio de Janeiro, Brazil, pp. 365-372, (2010)
- [Dij59] Edsger W. Dijkstra. A note on two problems in connexion with graphs. In: Numerische Mathematik. 1, S. 269–271, (1959)
- [Col87] C. Colbourn. The Combinatorics of Network Reliability, Oxford–University–Press, (1987)
- [Dod12] Markus Dod. Zuverlässigkeit in ungerichteten Netzwerken. Masterarbeit, (2012)
- [Doh98] Klaus Dohmen. Inclusion-Exclusion and Network Reliability. Electr. J. Comb., (1998)
- [Fus93] Donald S. Fussell, Vijaya Ramachandran, and Ramakrishna Thurimella. Finding Triconnected Components by Local Replacement. SIAM J. Comput. 22(3):587-616, (1993)
- [Gal05] J. Galtier, A. Laugier, und P. Pons. Algorithms to evaluate the reliability of a network. Design of Reliable Communication Networks, 2005. DRCN 2005. Proceedings.5th Int. Workshop on, Seite 8 pp., (2005)

- [Gou12] Luis Gouveia, Markus Leitner, and Ivana Ljubic. On the Hop Constrained Steiner Tree Problem with Multiple Root Nodes. ISCO, volume 7422 of Lecture Notes in Computer Science, page 201-212. Springer, (2012)
- [Hag83] Jane N. Hagstrom. Using the Decomposition-Tree of a Network in Reliability Computation. IEEE Transactions on Reliability, (Volume:R-32 , Issue: 1) page 71 - 78, (1983)
- [Har71] F. Harary. Graph Theory. Addison-Wesley, Reading-Mass, (1971)
- [Hop73] John E. Hopcroft, and Robert Endre Tarjan. Dividing a Graph into Tri-connected Components. SIAM J. Comput. 2(3):135-158, (1973)
- [Internet01] Vera Weil, Daniel Herrmann http://www.zaik.uni-koeln.de/AFS/teachings/ss09/ProgPrakt/Skript_Teil1.pdf, Aufrufam07.11.2013
- [Internet02] Thomas Grossmann http://www.gitta.info/Accessibility/en/html/Dijkstra_learningObject1.html#top, Aufrufam07.11.2013
- [Internet03] Alexander May http://www.cits.rub.de/imperia/md/content/may/dima08/09_bfs.pdf, Aufrufam11.11.2013
- [Jin11] R. Jin, L. Liu, B. Ding, H. Wang. Distance-Constrained Reachability Computation in Uncertain Graphs. Journal: Proceedings of the VLDB Endowment VLDB Endowment Homepage archive, Volume 4, Issue 9, Pages 551-562, (June 2011)
- [Luc96] Jacques Carlier, Corinne Lucet. A Decomposition Algorithm for Network Reliability Evaluation. Discrete Applied Mathematics 65(1-3): 141-156, (1996)
- [Luc97] Jean-Francois Manouvrier, Corinne Lucet. Resolving the Network Reliability Problem with a Tree Decomposition of the Graph. OPODIS: 193-204, (1997)
- [Luc00] Corinne Lucet, Jean-Francois Manouvrier, Jacques Carlier. Evaluating Network Reliability and 2-Edge-Connected Reliability in Linear Time for Bounded Pathwidth Graphs. Algorithmica 27(3): 316-336, (2000)
- [Luc07] Gary Hardy, Corinne Lucet, Nikolaos Limnios. K-Terminal Network Reliability Measures With Binary Decision Diagrams. IEEE Transactions on Reliability 56(3): 506-515 (2007)
- [Mal04] Jacek Malinowski. Non-binary decomposition trees - a method of reliability computation for systems with known minimal paths/cuts. Rel. Eng. and Sys. Safety 84(2):113-124, (2004)

- [Mig12] Denis A. Migov, Olga K. Rodionova. Calculating two-terminal reliability in a diameter constrained network with cutnodes. ICUIMC 2012: 130
- [Pet01] L. Petingi and J. Rodriguez. Reliability of networks with delay constraints. *Congressus Numerantium*, Vol. 152, pp. 117–123, (2001)
- [Pet08] L. Petingi. Combinatorial and Computational Properties of a Diameter Constrained Network Reliability Model. *APPLIED COMPUTING CONFERENCE (ACC '08)*, Istanbul, Turkey, May 27-30, (2008)
- [Pön01] Andre Pönitz, Peter Tittmann. Computing Network Reliability in Graphs of Restricted Pathwidth. (2001)
- [Rob83] N. Robertson and P.D. Seymour. Graph minors. I. Excluding a forest. *J. Comb. Theory Series B*, 35, 33-61, (1983)
- [Rob86] N. Robertson and P.D. Seymour. Graph minors. II. Algorithmic aspects of treewidth. *J. Algorithms*, 7, 309-322, (1986)
- [Sat96] Appajosyula Satyanarayana, R. Kevin Wood, Leonidas Camarinopoulos, and G. Pampoukis. Note on A Linear-Time Algorithm for Computing K-Terminal Reliability in a Series-Parallel Network. *SIAM J. Comput.* 25(2):290, (1996)
- [She91] D.-R. Shier. Network reliability and algebraic structure. Clarendon-Press, Oxford, (1991)
- [Tan07] Christian Tanguy. What is the probability of connecting two points? *J. Phys. A: Math. Theor.* 40, 14099–14116, (2007)
- [Tar72] Robert Endre Tarjan. Depth-First Search and Linear Graph Algorithms. *SIAM J. Comput.* 1(2):146-160, (1972)
- [Val79] L.-G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, Vol. 8, pp. 410–421, (1979)

Erklärung

Hiermit erkläre ich, dass ich meine Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die Arbeit noch nicht anderweitig für Prüfungszwecke vorgelegt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Mittweida, 23.11.2013